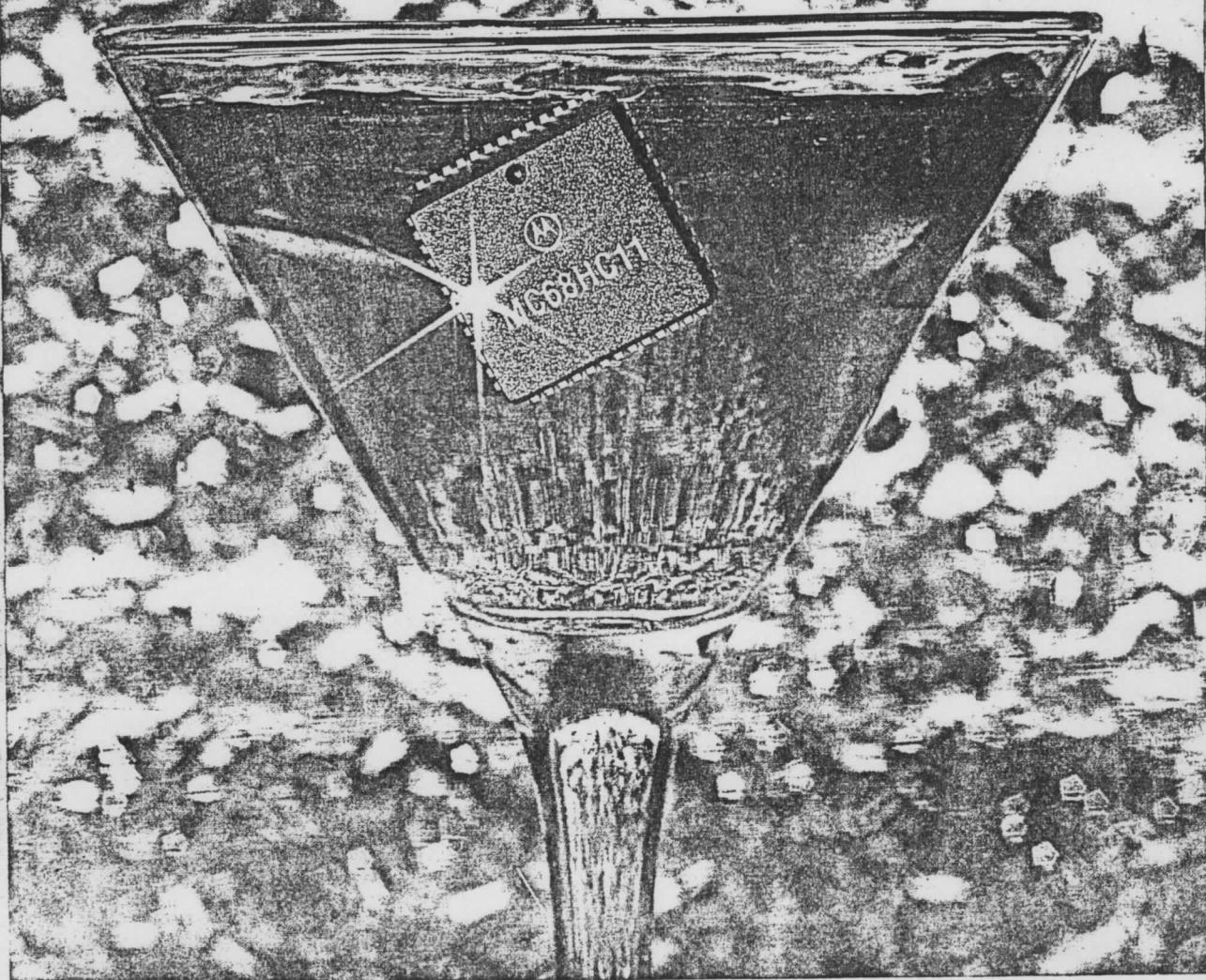


MC68HC11

SINGLE CHIP MICROCOMPUTER

cool and powerful



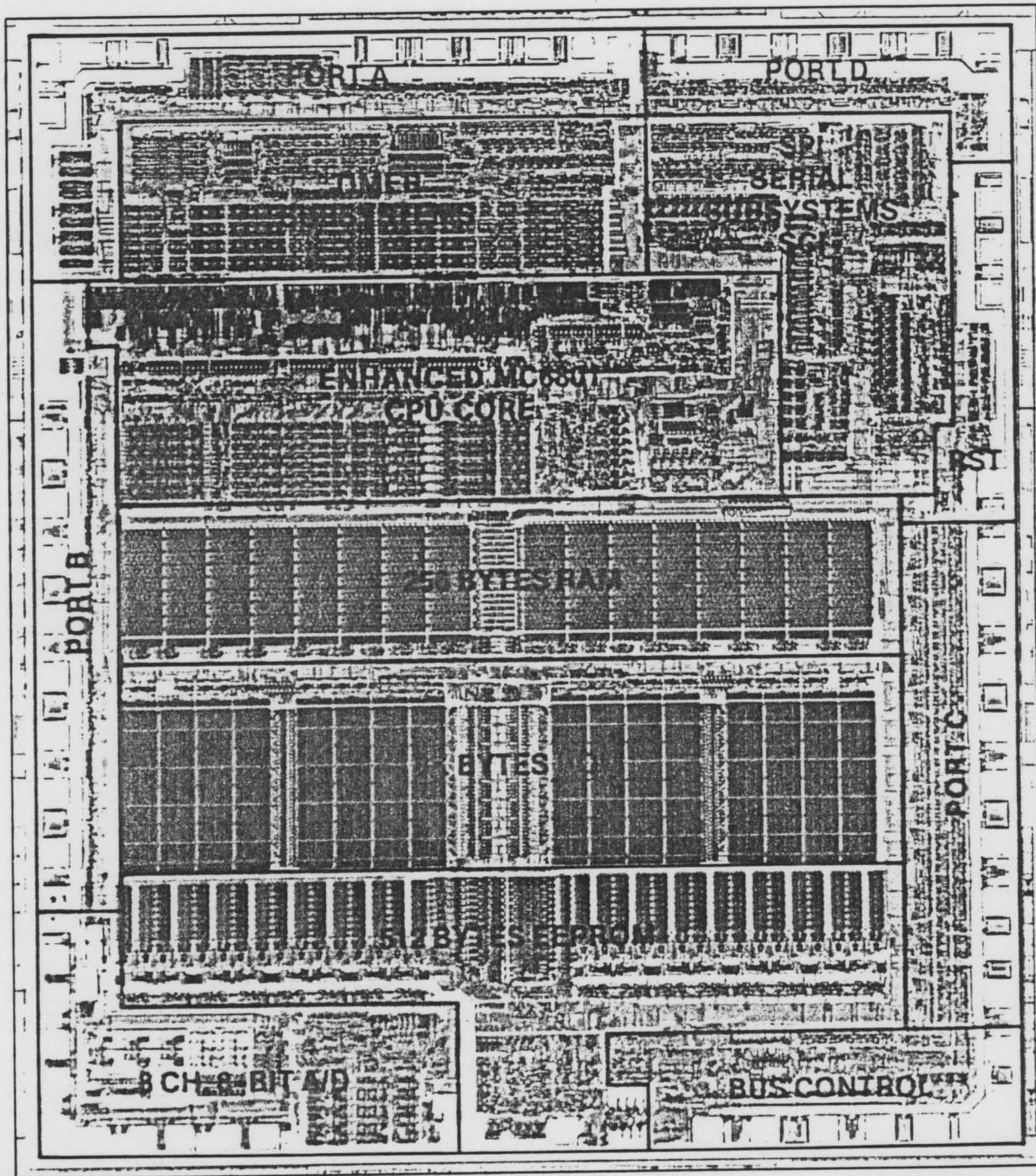
MOTOROLA

SILVERSTAR

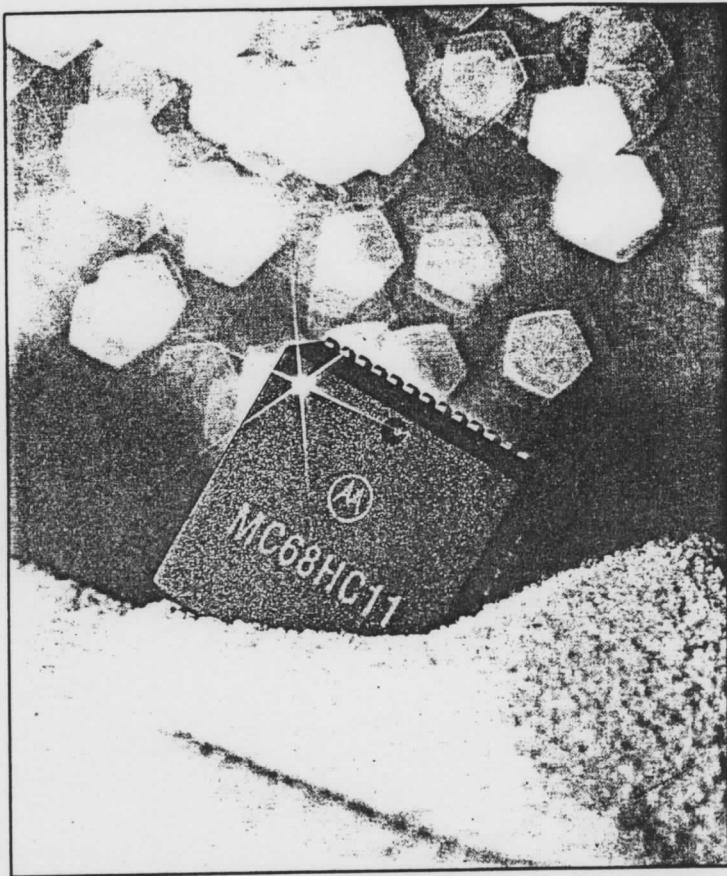
- 20146 Milano - Via dei Gracchi, 20 - Tel. (02) 4996 - Fax (02) 435594 - Telex 332189 SIL MI
- 00198 Roma - Via Paisiello, 30 - Tel. (06) 8848841 - Fax (06) 863228 - Tel. 610511
- 10139 Torino - P. Adriano, 9 - Tel. (011) 443275/6 - 442321 - Fax (011) 4473306 - Tel. 220181
- 40122 Bologna - Via Del Porto, 30 - Tel. (051) 522231 - Fax (051) 552563
- 35010 Cadoneghe (PD) - Vittorio Emanuele, 2/6 - Tel. (049) 706533 - Fax (049) 703376
- 50133 Firenze - Via Masaccio, 175 - Tel. (055) 572418 - Fax (055) 579575

THE MC68HC11 MICROCOMPUTER

Flexibility, in the world of single chip microcomputers, is a measure of how well a particular device can be moulded to any application. Functionality is the means of achieving this. With the MC68HC11, Motorola has created the ultimate microcomputer for flexibility through functionality.



The MC68HC11A8 Die.



MC68HC11 Plastic Quad Package.

CONSIDER THE FACTS!

On-Chip EEPROM

The enhanced MC6801 8-bit CPU is supported by 3K bytes of ROM, 256 bytes of RAM and the flexibility of 512 bytes of EEPROM. Single power rail operation is achieved by the use of an on-chip charge pump. Individual devices can be personalised, either during production or in real time, and the EEPROM has a security feature to prevent unauthorised reading of its contents. Furthermore, if the on-board memory does not suit your requirements, an expansion bus allows direct access of up to 64K bytes of off-board memory using 18 I/O lines.

Interrupts

Twenty-one interrupt sources, including resets, each have their own vector to facilitate good prioritisation and rapid servicing, and prevent important events being missed.

Timer Structure

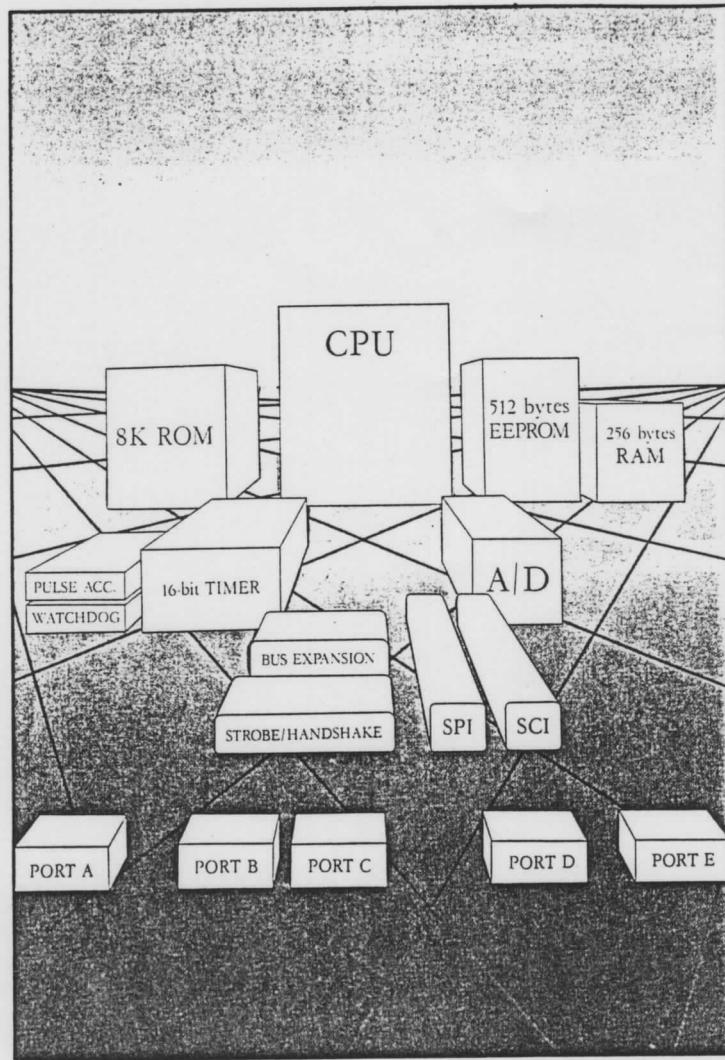
A sophisticated timer structure includes a 16-bit free-running counter, with a prescaler and a suite of input capture and output compare registers.

Other features include an 8-bit Pulse Accumulator and a Real Time Interrupt. System protection is enhanced with the inclusion of a Computer Operating Properly (COP) Watchdog System, and a Clock Monitor.

Parallel and Serial I/O

Interfacing with the outside world is another strength of the MC68HC11. Forty I/O lines are complemented by parallel handshaking capability on two ports. Serial communications are provided, both synchronously and asynchronously, through two independent communications channels.

The Serial Communications Interface (SCI) is an asynchronous NRZ format channel allowing transmission rates from 75Hz to 131KHz, with built-in data recovery/noise immunity using an advanced technique of majority sampling. The serial peripheral interface (SPI) is a four wire synchronous system allowing rates up to 1.05 MHz in multi-master systems. It can be configured as a three-wire system if total multi-master flexibility is not required.



MC68HC11A8 Block Diagram.

A/D Conversion

Completing the hardware functionality is an 8-channel, 8-bit accuracy A/D converter to allow direct connection to the typically analogue external environment. Its implementation as a switched capacitor network ensures accuracy over the full operating temperature range. Additionally, an on-chip RC network allows greater accuracy below 1 MHz bus frequency.

Expanded Instruction Set

An extra index register, the addition of 16 by 16 fractional and integer divide instructions, and powerful bit manipulation capability have added 91 new instructions to the industry standard M6801 instruction set, making 311 instructions in total. An illegal opcode detection circuit is included for increased system security.

HCMOS Technology

The MC68HC11A8 is the implementation of all these features in 2 micron, high speed, high

density CMOS (HCMOS) on one single piece of silicon. The use of HCMOS allows operation at double the bus speed of its HMOS predecessors as well as drastically reducing power consumption, improving noise immunity and allowing operation up to 125°C.

It's cool and powerful – the MC68HC11A8.

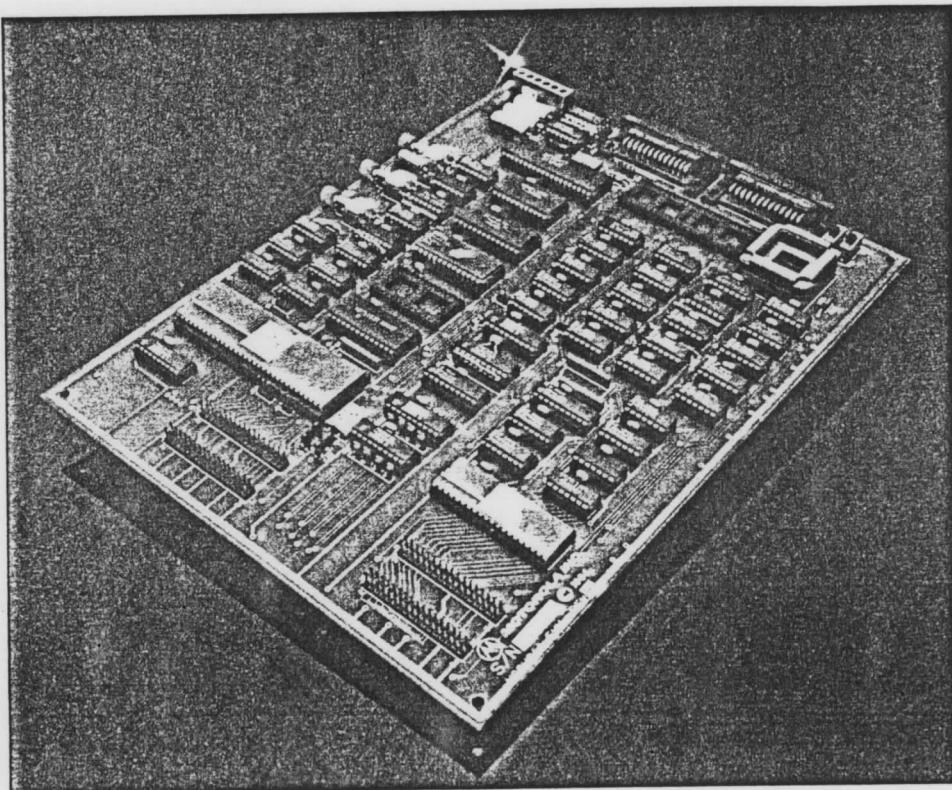
Future Developments

The MC68HC11A8 is the first product in the M68HC11 family. Future devices will include the MC68HC811A2 and MC68HC811A8, with no ROM, but with 2K and 8.5K of EEPROM respectively.

Packaging

The MC68HC11A8 is available in a 52-pin plastic quad pack, or a 48-pin DIL package. The 48 pin DIL package does not have A/D channels 4 to 7 bonded out; consequently only 4 analogue to digital channels are available.

SYSTEMS SUPPORT FOR THE MC68HC11



MC68HC11 Evaluation Module (EVM).

MC68HC24

The MC68HC24 is a port expansion device which replaces Ports B and C and the parallel handshaking of the MC68HC11 when the MCU is used in expanded mode. Applications requiring external memory in early production can use the MC68HC24 to address this memory without losing Ports B and C. This part, in conjunction with an MC68HC11, a CMOS EPROM and two CMOS gates allows simple construction of a five chip prototyping circuit. An expanded system using the MC68HC24 is software and hardware compatible to the MC68HC11 in both single chip mode and special test mode.

In addition, other CMOS microcomputer systems requiring I/O expansion or a parallel printer interface can use the MC68HC24 as a cost effective solution. It is not restricted to replacing MC68HC11 ports. The MC68HC24 is available in 44-pin plastic quad or 40-pin DIL packages.

Motorola has in development the MC68HC811A8 prototyping part. The 8K ROM of the MC68HC11A8 will be replaced by 8K of EEPROM, and exact emulation of the ROM-based part will be possible for portable or space critical applications.

MC68HC11 Evaluation Module

The MC68HC11 Evaluation Module (EVM) provides a low cost method for extensive evaluation and preliminary development work. It is not intended as a replacement for the high-powered emulator system, described later, but it does provide many of the features associated with such systems.

Full emulation of the MC68HC11 is provided in both single chip and expanded modes at full speed (2.1 MHz). The EVM runs with dual memory maps to allow access to the full 64K of addressable memory. CMOS RAM is used to emulate the MC68HC11's on-chip ROM and EEPROM – this RAM is expandable to allow for future family members.

RS232C serial communication is provided for a host and/or dumb terminal, enabling software to be downloaded into the EVM. Finally the on-board monitor enables debugging of software, and single line assembly/disassembly is provided.

SYSTEMS SUPPORT FOR THE MC68HC11

MC68HC11 DEVELOPMENT SYSTEMS SUPPORT

Together with the new MC68HC11 family of MCUs, Motorola introduces its new HDS-300 development system.

Complete MC68HC11 system development requires the use of sophisticated hardware and software to ensure speedy and error-free results. Motorola's offering is both flexible and comprehensive.

HARDWARE

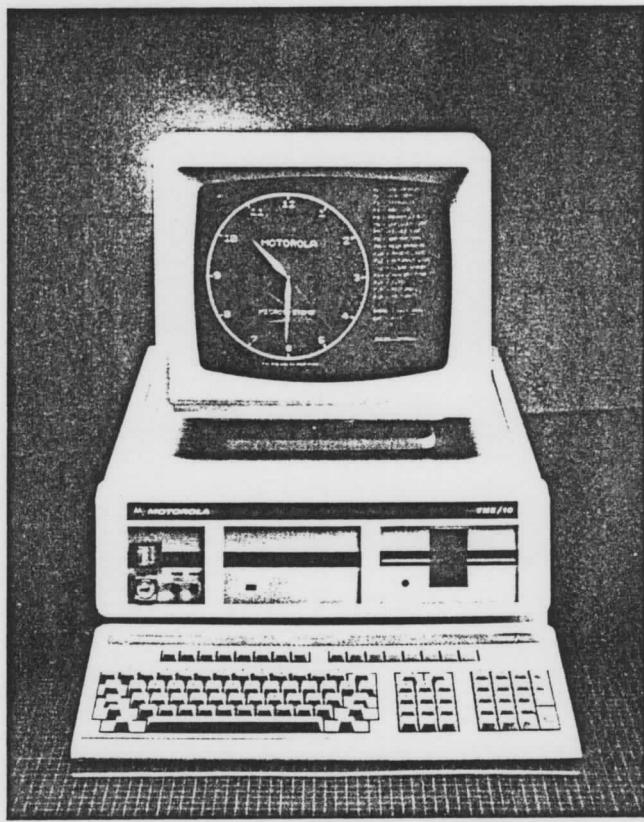
1. *Development computer for software development, storage and documentation: VME/10, EXORMacs or any host capable of down-loading Motorola S Records and running an MC68HC11 cross assembler.*
2. *Hardware development station for hardware development, real time emulation and logic signal analysis: HDS-300.*

SOFTWARE

1. *Structured relocatable Macro-Assembler.*
2. *C-Compiler producing very byte-efficient MC68HC11 code (only for System V/68).*
3. *Powerful linker.*
4. *Convenient screen oriented editor.*
5. *Communications Software.*
6. *Symbolic Debug.*

The above software is available to run under VERSAdos or Motorola's System V/68. System V/68 is derived from AT & T Bell Laboratories' UNIX System V.

VME/10



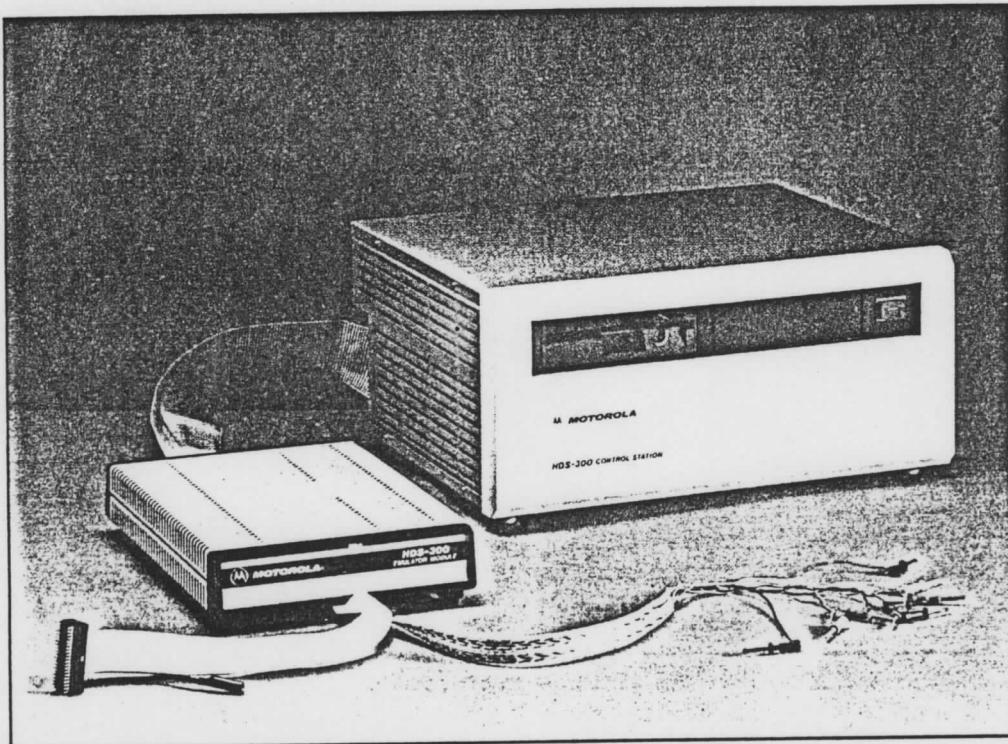
VME/10 Development Computer.

The VME/10 is the ideal development computer for MC68HC11 software since it also offers a growth path to even higher performance. Two powerful operating systems are available for the VME/10: realtime VERSAdos and Motorola's System V/68. Both operating systems carry a full set of development software.

The VME/10 contains the powerful MC68010 16-bit microprocessor, and supports all 8, 16 and 32-bit development work. All assemblers on the VME/10 accept structured commands. Winchester and floppy disk drive storage is provided and there is 384KB of RAM. A VMEbus card cage allows the VME/10 to be expanded.

SYSTEMS SUPPORT FOR THE MC68HC11

HDS-300



HDS-300 Control Station with Emulator Module.

The HDS-300 hardware development station consists of:

- *The HDS-300 main control station*
- *The processor dependent personality module*
- *A set of 5.25" disks containing control software*

Personality modules will be available for M68HC11, M6809 and M6801 microprocessor families. The HDS-300 interfaces (via two RS232C serial links) to a host computer and a terminal. No extra terminal is required when used with a VME/10. It also allows a printer with a Centronics interface to be connected directly.

The HDS-300 enables you to:

- *Load MC68HC11 software from the development computer.*
- *Run this software under real time conditions.*
- *Debug and modify this software.*
- *Store debugged software on the HDS-300 built-in floppy disk drive.*

Software debugging can be carried out either in the prototype hardware, where the processor is substituted by the emulator, or in the HDS-300 in stand-alone mode. When connected to the prototype hardware, the HDS-300 provides the

following facilities to assist hardware debugging:

- *Check functionality of RAM, ROM and I/O devices.*
- *Investigate bus activities.*
- *Unfold complex processor internal processes.*

The key features for the user are:

- *Convenient and informative handling.*
- *Multi-level Help, usable as an on-line manual.*
- *Host and terminal independent.*
- *Local symbol table which enables full symbolic debug.*
- *Line assembler/disassembler.*
- *Command macros, macro edit and macro storage.*
- *High speed emulation RAM, 32 Kbyte standard or 64 Kbyte optional.*
- *Realtime trace with 1K x 64 bit trace buffer. This can trace all processor signals plus user defined signals.*
- *Disassembly of trace buffer.*
- *Histogram performance analysis. Especially useful for investigation of multi-tasking software.*
- *Multi-processor support.*
- *Synchronisation I/O to trigger logic analyser, oscilloscopes or other HDS-300's.*
- *Future option for high level language symbolic debug.*

APPLICATIONS FOR THE MC68HC11A8

Many applications for the MC68HC11A8 already exist in the Telecommunications, Automotive and Industrial markets, in areas as diverse as PABX, Anti-Skid Braking and Data Logging. Each application utilises one or more of the innovative features to give the final product a competitive edge.

The following specific applications illustrate the advantages of some of the advanced features of the MC68HC11A8.

MC68HC11A8 Engine Management System

The introduction of more stringent exhaust emission regulations, and the requirement for precise fuel and ignition controls on internal combustion engines, make electronic engine management systems essential. This application demonstrates that by the careful choice of the correct microcomputer a comprehensive engine management system can be built simply and at low cost. The timer functions, on-chip A/D converters and HCMOS technology make the MC68HC11A8 the ideal device.

The application shows a multi-toothed wheel system, where the wheel is attached to the crank-shaft and a missing tooth generates a reference point. The signals generated from the toothed wheel indicate crankshaft position and timing information for ignition coil control, fuel injection and the idle speed control valve.

The analogue signals – manifold pressure, inlet air temperature and engine coolant temperature – are fed into the on-chip A/D converters on the MC68HC11. A further analogue channel reads the band-pass filtered integrated knock signal to provide a closed loop ignition system.

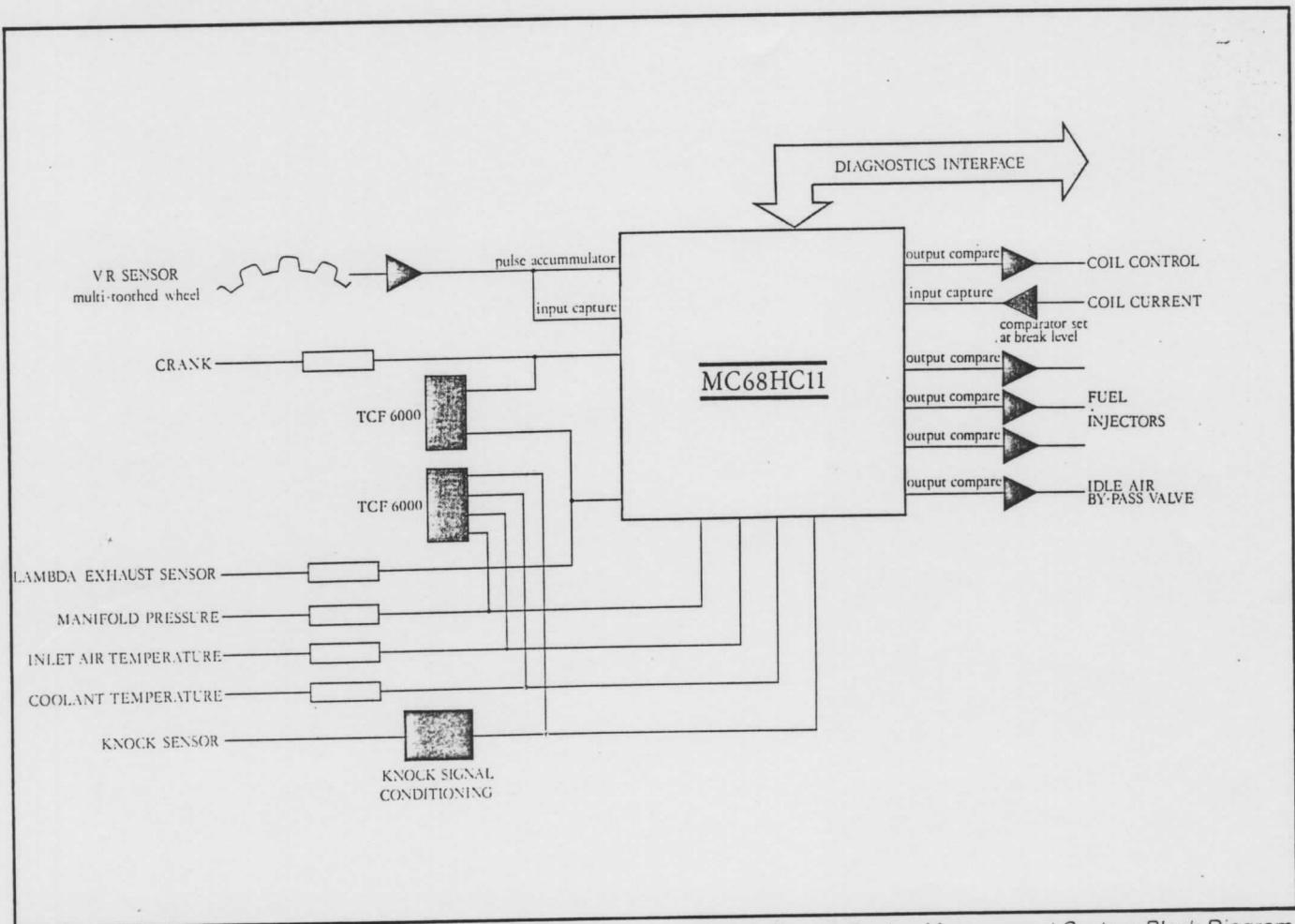
The fifth analogue channel is used for reading the zirconium dioxide sensor in the exhaust, which indicates excess oxygen in the exhaust gas; this provides closed loop fuel control.

System Operation

The signal picked up from the multi-toothed wheel is used to drive the pulse accumulator as a tooth counter, and an input capture register to measure the time between pulses. The missing tooth is detected by comparing the time periods between consecutive teeth. The leading edge of the next tooth is used as a reference mark and all timing calculations are made relative to this point. The 16-bit free running timer is read at every reference mark and hence engine speed can be determined.

Ignition Control

The measured speed, engine load and engine temperature are entry parameters for a three dimensionally mapped spark advance table. The resultant period from reference point to spark is timed precisely using the pulse accumulator to count whole teeth, and when it overflows, using an output compare register to provide a fractional tooth time. The spark coil is turned on by the hardware autotoggle feature, and turned off by current level sensing or by using the output compare function to generate a pre-calculated time period. Optimal spark strength and timing are obtained by closed loop feedback using the coil current level.



MC68HC11A8 Engine Management System Block Diagram.

Fuel Control

The Fuel On period, in terms of crankshaft degrees, is derived from a complex calculation involving mass air and fuel flow. This calculation is greatly simplified using the hardware multiply and divide instructions of the MC68HC11.

By storing the calculated fuel time in the output compare registers the injectors can be timed to a resolution of 1 tooth. Further resolution can be obtained by using a proportion of the time between teeth as in the ignition control.

The output lines can be toggled automatically by the internal hardware on a valid output compare. This reduces software overheads and significantly improves timing accuracy. This hardware toggle feature can also be used effectively in an idle

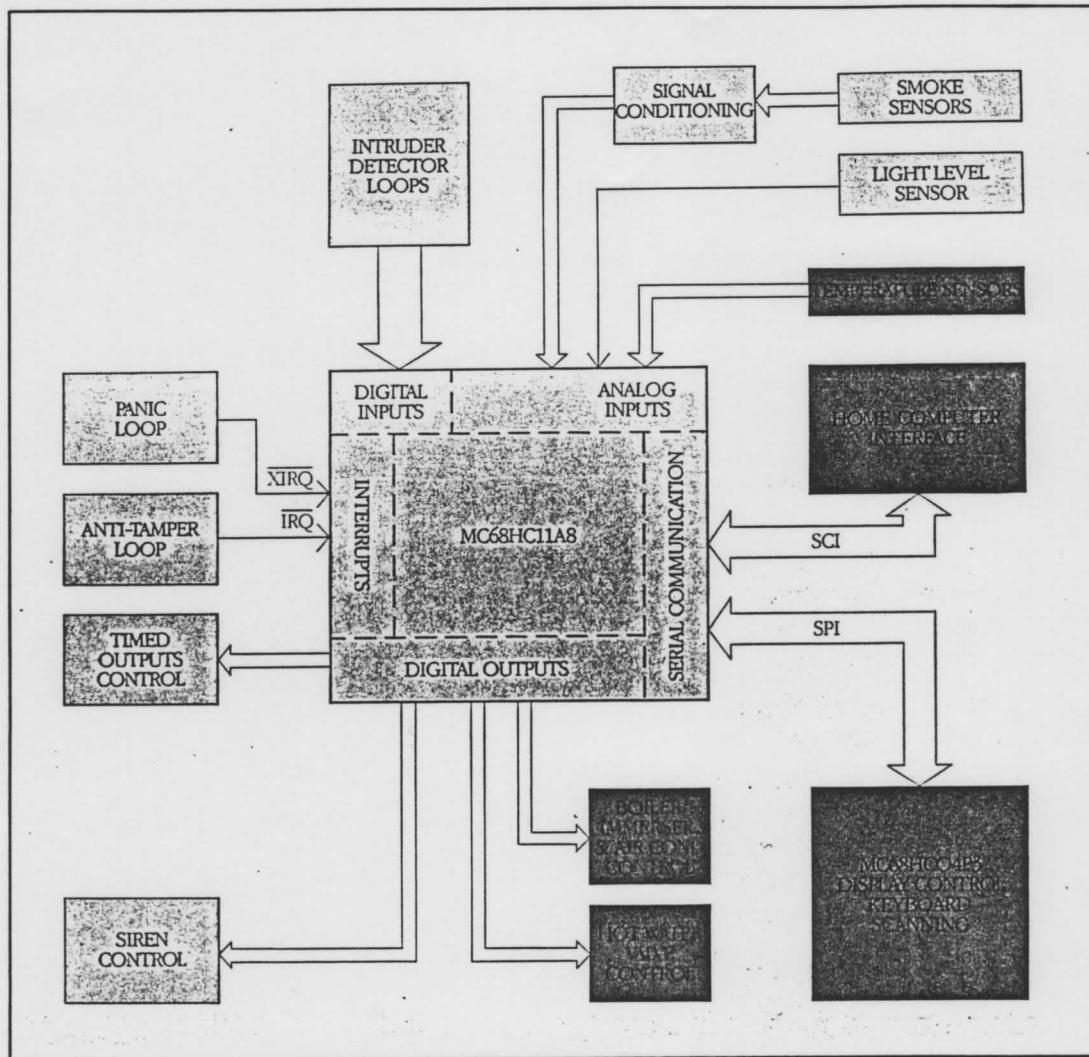
speed controller which uses pulse width modulation to control a valve. The pulse width modulation can be achieved simply by storing the desired toggle point in an output compare register – the MC68HC11 hardware drives the valve automatically.

So the complex process of engine management is achieved simply and effectively by utilising the powerful timer structure of the MC68HC11A8, and monitoring environmental conditions through the A/D channels.

In this application the noise immunity and wide operating temperature range of HCMOS are extremely valuable.

APPLICATIONS FOR THE MC68HC11A8

MC68HC11A8 Home Control System



MC68HC11A8 Home Control System Block Diagram.

There are many applications in the home that are suitable for MCU control. In this application concept, Home Security and Environmental Control have been integrated into one system. With the increasing cost of energy for heating and cooling, an MCU based system is ideal for minimising the energy consumed in maintaining the required hot water and air temperatures at various times of the day. The cost of running the heating system is monitored and the impact of various heating and cooling strategies can be observed.

Rising crime trends make a home security system a necessity to reduce the cost of house insurance and to provide peace of mind. This system provides both fire and intrusion warning.

Efficient use of an MC68HC11A8 as the main controller minimises the external hardware required due to the high level of integration on the

chip. The system makes use of the following key features: EEPROM, A/D, SPI, SCI and the Real Time Clock (periodic interrupt). These advanced features allow the design of a powerful, low cost system. HCMOS technology plus the watchdog timer, clock monitor and illegal op code trap allow the production of a very secure system with high noise immunity.

General System Features

The system consists of a control box, containing the MC68HC11A8 and related switching and sensing hardware, with a remote keypad and display unit using an MC68HC04P3 communicating via the SPI. The MC68HC04P3 SPI is implemented in software. For security purposes the system has battery back up.

A clock is derived from the Periodic Interrupt and this is used as a timer to control various equip-

APPLICATIONS FOR THE MC68HC11A3

ment. The timer works on a weekly cycle to allow for different daily requirements – more heating at the weekend for example.

Using the SCI, a home computer can be linked to the system to allow data acquisition and control capability if required.

Configuration data and timer information for the environmental and security systems is stored in the EEPROM. The configuration data is set up initially by a computer connected to the SCI, or from the keypad.

Environmental Control

Temperature and time inputs are used by the environmental system to control the central heating, air conditioning and hot water.

Temperature Inputs

There are four temperature sensors connected to A/D inputs. One measures the outside air temperature and two measure the internal temperatures in two areas. The remaining sensor measures the hot water temperature. These inputs are combined with information set up on the system timer to regulate the rest of the system.

Continuously measuring the rate of change of temperature internally and externally, and applying that data to a pre-defined mathematical model of the system, makes predictive control possible. The powerful 16 bit mathematical instructions simplify this process. Furthermore, by storing some of the parameters of the model in EEPROM, the system may also be made adaptive. Any over or undershoot of targets would cause the supervisor program to modify the parameters in an attempt to create a more accurate model and so reduce the error rate.

Water Control Valves

There are two temperature measurements inside, and they can be used to control two separate heating zones. The hot water from the boiler is directed to the heating zones and/or hot water cylinder by control valves.

Cost Monitoring

Based on the cost per hour of running the boiler, air conditioning and immersion, and the time each is used, it is possible for the system to calculate the cost of controlling the house environment. This information is stored in EEPROM and can be displayed on the keypad/display unit or on a home computer.

Home Security

The home security system consists of a fire alarm and an intruder alarm.

1. Fire Alarm

Two smoke sensors and the two interior air temperature sensors are monitored to detect a fire. Analogue to digital inputs are used for these sensors. The system is programmed not to switch on the air conditioning if the interior temperature rise is very rapid, as the air conditioning would supply oxygen to any combustion. When the system has sensed a rapid rise in temperature or smoke the sirens are triggered, and the fire warning is switched on and off repeatedly to distinguish the sound from the continuous siren of the intrusion alarm.

2. Intrusion Alarm

A variety of detectors provide the inputs to the security system. These could include:-

Space Protection by Infra-red, microwave or ultrasonic scanners

Door switches

Pressure pads

Window vibration sensors

Each loop would cover a zone or type of intruder detector. The EEPROM contains information on the zone name or the type of detector. In addition individual loops can be disabled if, for example, a door has to be left open that would normally trigger the alarm. Entry and exit time delays are also stored in the EEPROM. There are two sets of configured information: one set is used when the building is occupied, the other when it is empty. Certain sensor loops may have to be disabled when the building is occupied at night, and likewise different delay times may be required.

The anti-tamper loop and the panic button loop are connected to interrupts. The anti-tamper loop passes through all parts of the system and if broken will immediately trigger the alarm. The panic button loop switches are located at external doors, and again the alarm is immediately triggered. Both these loops remain active when the rest of the alarm system is switched off.

Four timed outputs are provided to switch on lights, etc., at pre-programmed times. One of the A/D inputs has a light level sensor attached, and this can also be used to trigger the timed outputs.



MOTOROLA

Zilogix Quad
linear -
Dinchi.

270.000

81PZ DIL

AAFN + \$26.000 (13x 4 lines)
A1P
AOP

MC68HC11A4

A6
A1
A8

Product Preview

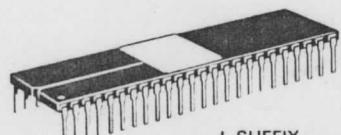
MC68HC11A4 8-BIT MICROCOMPUTER

The HCMOS MC68HC11A4 is an advanced microcomputer (MCU) containing highly sophisticated on-chip peripheral functions. An improved instruction set provides additional capability while maintaining compatibility with the other members of the M6801 Family. The fully static design allows operation at frequencies down to dc, further reducing its already low power consumption. Features available in addition to the normal M6801 features include:

- 4K Bytes of ROM
- 512 Bytes of EEPROM
- 256 Bytes of RAM (All Saved During Standby)
- Enhanced 16 Bit Timer System
 - Four Stage Programmable Prescaler
 - Three Input Capture Functions
 - Five Output Compare Functions
- An 8-Bit Pulse Accumulator Circuit
- An Enhanced Non-Return-To-Zero Serial Communications Interface (SCI)
- A New Serial Peripheral Interface
- Eight Channel 8-Bit A/D Converter
- A Real Time Interrupt Circuit
- A Computer Operating Properly (COP) Watchdog System

HCMOS
(HIGH-DENSITY HIGH-PERFORMANCE
SILICON GATE)

MICROCOMPUTER

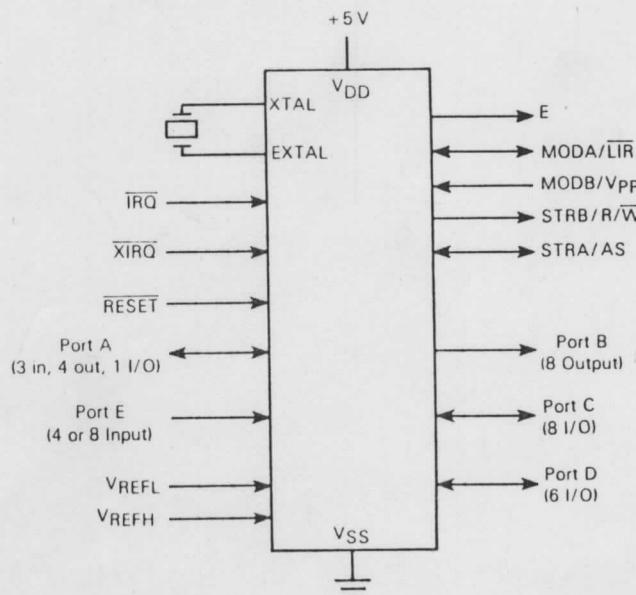


L SUFFIX
CERAMIC PACKAGE

Also Available In
52 Pin Quad Pack

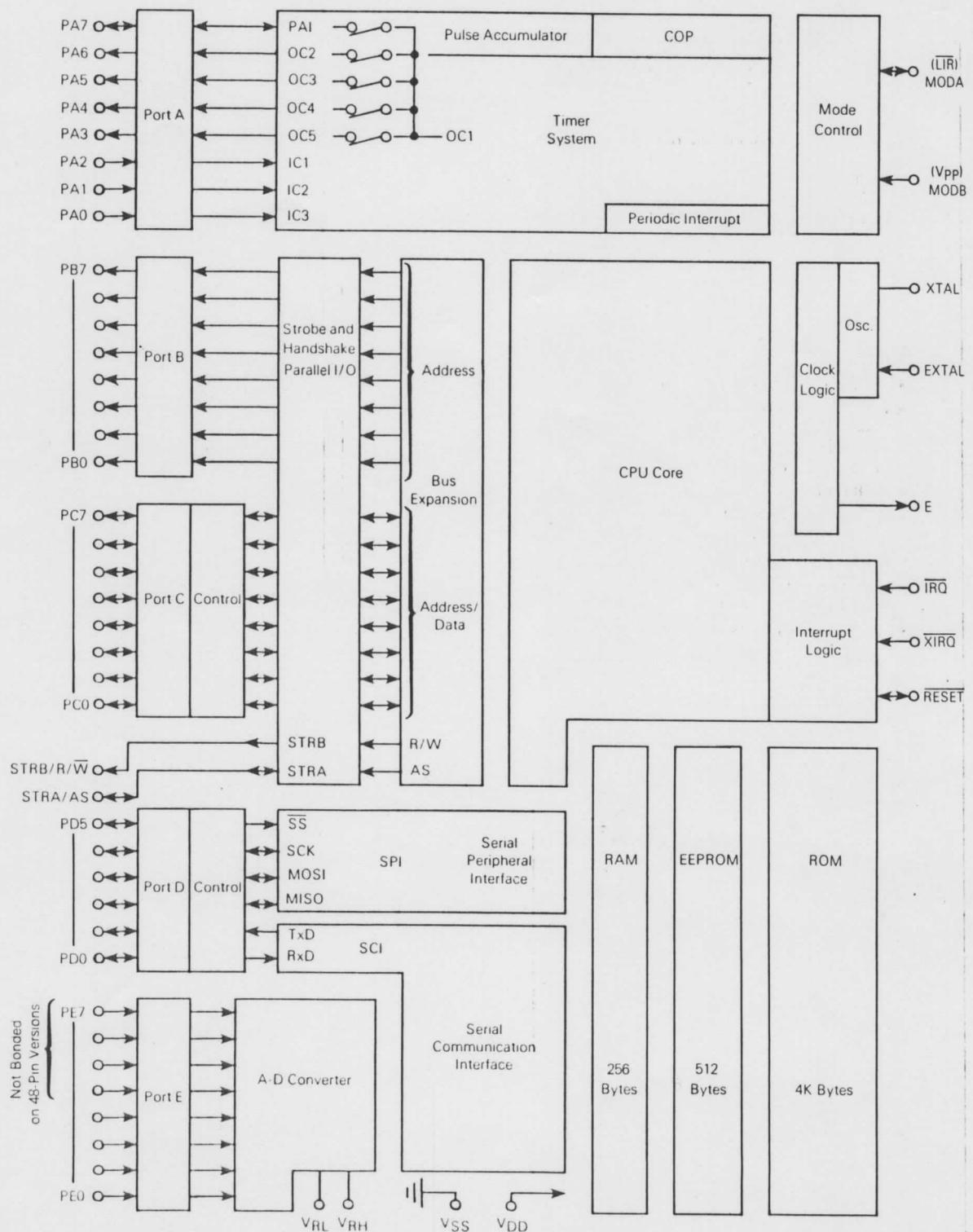
3

PIN ASSIGNMENT



MC68HC11A4

FIGURE 1 — MC68HC11A4 BLOCK DIAGRAM



MC68HC11A4

GENERAL DESCRIPTION

The MC68HC11A4 is a single-chip microcomputer that utilizes HCMOS techniques to provide the low power characteristics and high noise immunity of CMOS plus the high-speed operation of HMOS. On chip memory systems include a 4K byte ROM, 512 bytes of electrically erasable programmable ROM (EEPROM), and 256 bytes of static RAM. The MC68HC11A4 microcomputer also provides highly sophisticated, on-chip peripheral functions including an 8-channel analog-to-digital converter, a serial communications interface (SCI) subsystem, and a serial peripheral interface (SPI) subsystem.

New design techniques are used to provide a 2 MHz nominal bus rate. The timer system is expanded to provide three input capture lines, five output compare lines, and a real time interrupt circuit. This gives the MC68HC11A4 one of the most comprehensive timer systems found on a single-chip microcomputer. Other features of the MC68HC11A4 include: a pulse accumulator which can be used to count external events (event counting mode) or measure an external period (input gates accumulation of internal clock - E/64), a computer operating properly (COP) watchdog system which helps protect against software failures, a programmable clock monitor system which causes generation of a system reset in case the clock is lost or running too slow, and an illegal opcode detection circuit which provides an unmaskable interrupt if an illegal opcode fetch is detected.

OPERATING MODES

The MC68HC11A4 MCU uses two dedicated pins (MODA and MODB) to select one of two basic operating modes or one of two special operating modes. The basic operating modes are single chip (mode 0) and expanded multiplexed (mode 1), and the special operating modes are bootstrap and special test. The levels required on the MODA and MODB pins for mode selection are discussed in **FUNCTIONAL PIN DESCRIPTION**. The characteristics of the operating modes are discussed below.

SINGLE-CHIP MODE (MODE 0)

In the single-chip mode the MCU functions as a self-contained microcomputer and has no external address or data bus. In this mode the MCU provides maximum use of its pins for on-chip peripheral functions, and all address and data activity occurs within the MCU. As discussed in **FUNCTIONAL PIN DESCRIPTION**, when MODA = 0 and MODB = V_{DD} the single chip mode is selected during reset.

EXPANDED MULTIPLEXED MODE (MODE 1)

In this mode, two I/O ports plus two additional I/O lines become address, data, and control (AS and R/W) to allow the MCU to address up to 64K bytes of address space. High order address bits are output on the port B pins. Low order address bits and the data bus are time multiplexed on the eight port C pins. Port D bit 6 becomes the address strobe (AS) control output which is used in demultiplexing the low order address from the data at port C. The R/W control pin (port D, pin 7) is used to control the direction of data transfers on the port C bus. Refer to **FUNCTIONAL PIN**

DESCRIPTION and INPUT/OUTPUT PORTS for additional information regarding address strobe, read/write, port B, and port C.

BOOTSTRAP MODE

The bootstrap mode is considered a special mode as distinguished from the normal operating single-chip mode. In the bootstrap mode, all vectors are fetched from the 128 byte on-chip boot loader ROM. This is a very versatile mode since there are essentially no limitations on the special purpose program that is boot loaded into the internal RAM. The boot loader is contained in 128 bytes of ROM which is enabled as internal memory space at \$BF80-\$BFFF. The boot loader contains a small program which reads a 256 byte program into on-chip RAM (\$0000-\$00FF). After the character for address \$00FF is received, control is automatically passed to that program at memory address \$0000 and the MCU operates in the single chip mode.

In the bootstrap mode, the serial receive logic is initialized by software in the boot loader ROM. This allows the program control of the serial communications interface (SCI) baud and word format.

During initialization of the special bootstrap mode, a special control bit is configured to permit access to a number of special test control bits which allows for self testing of the MCU in the bootstrap mode. Also, since the mode control bits can be written to, the operating mode of the MCU may be changed from the special bootstrap mode (which is a single-chip mode by default) to expanded multiplexed mode under program control.

TEST MODE

The test mode is considered a special mode as distinguished from the normal operating expanded multiplexed mode; however, it is considered as operating in the expanded multiplexed mode since external memory may be addressed. The reset vector is fetched from external memory space \$BFFE-\$BFFF; therefore, program control may be vectored to an external test program.

The test mode is primarily intended as the main production test mode at the time of manufacture; however, it may also be used to program calibration or personality data into the internal EEPROM (electrically erasable programmable read only memory) of the MC68HC11A4. During initialization of the test mode, a special control bit is configured to permit access to a number of special test control bits. Also, since the mode control bits can be written to in the test mode, the operating mode of the MCU may be changed from the special test mode (which is an expanded multiplexed mode by default) to the single-chip mode under program control.

MEMORY

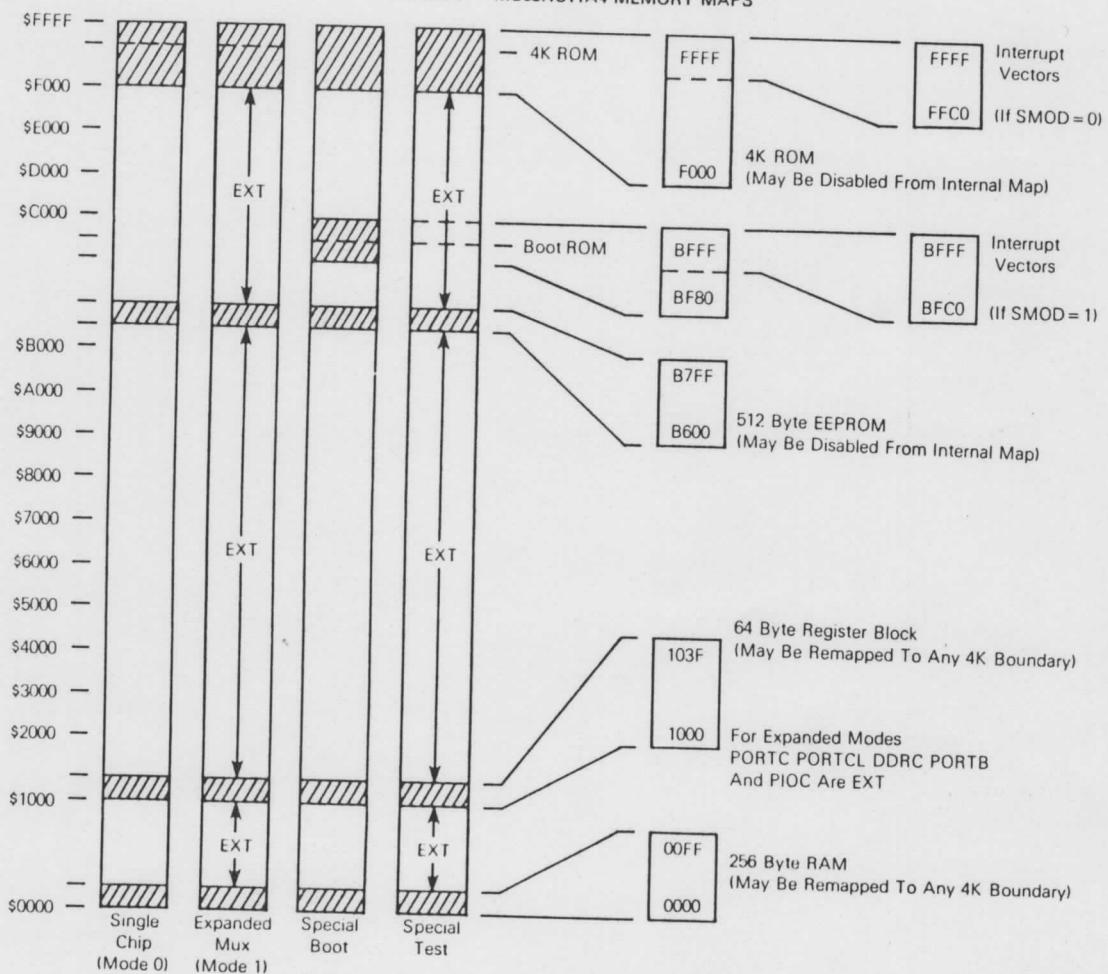
Composite memory maps for each MC68HC11A4 mode of operation are shown in Figure 2. These modes include single-chip, expanded multiplexed, special boot, and special test.

In the single-chip mode (mode 0) of Figure 2, the MC68HC11A4 does not generate external addresses. The actual internal memory locations are shown in the shaded areas of Figure 2 and the contents of these shaded areas are

MC68HC11A4

3

FIGURE 2 — MC68HC11A4 MEMORY MAPS



shown on the right side of the diagram. For example: memory locations \$0000 through \$00FF contain the 256 bytes allocated to RAM; memory locations \$1000 through \$103F are allocated for a 64 byte register block; memory locations \$B600 through \$B7FF are allocated for a 512-byte EEPROM (electrically erasable programmable read only memory); and memory locations \$F000 through \$FFFF are allocated for 4K bytes of ROM (memory locations \$FFC0 through \$FFFF are reserved for the interrupt and reset vectors).

The expanded multiplexed mode (mode 1) memory locations shown in Figure 2 are basically the same as for the single-chip mode; however, the memory locations between the shaded areas (designated EXT) are for externally addressed memory and I/O.

The special bootstrap mode memory locations are similar to the single chip memory locations except that a special

bootstrap program is addressed at memory locations \$BF80 through \$BFFF. The special bootstrap program controls the process of boot loading a 256 byte program through a serial port into internal RAM.

The special test mode memory locations are similar to the expanded multiplex mode except the interrupt vectors are at external memory locations.

CENTRAL PROCESSING UNIT

The central processing unit (CPU) of the MC68HC11A4 is basically an extension of the MC6801 CPU. In addition to being able to execute all MC6800 and MC6801 instructions, the MC68HC11A4 uses a 4-page opcode map to allow execution of 91 new opcodes. As in the MC6801, the CPU of the MC68HC11A4 is implemented independently from the I/O,

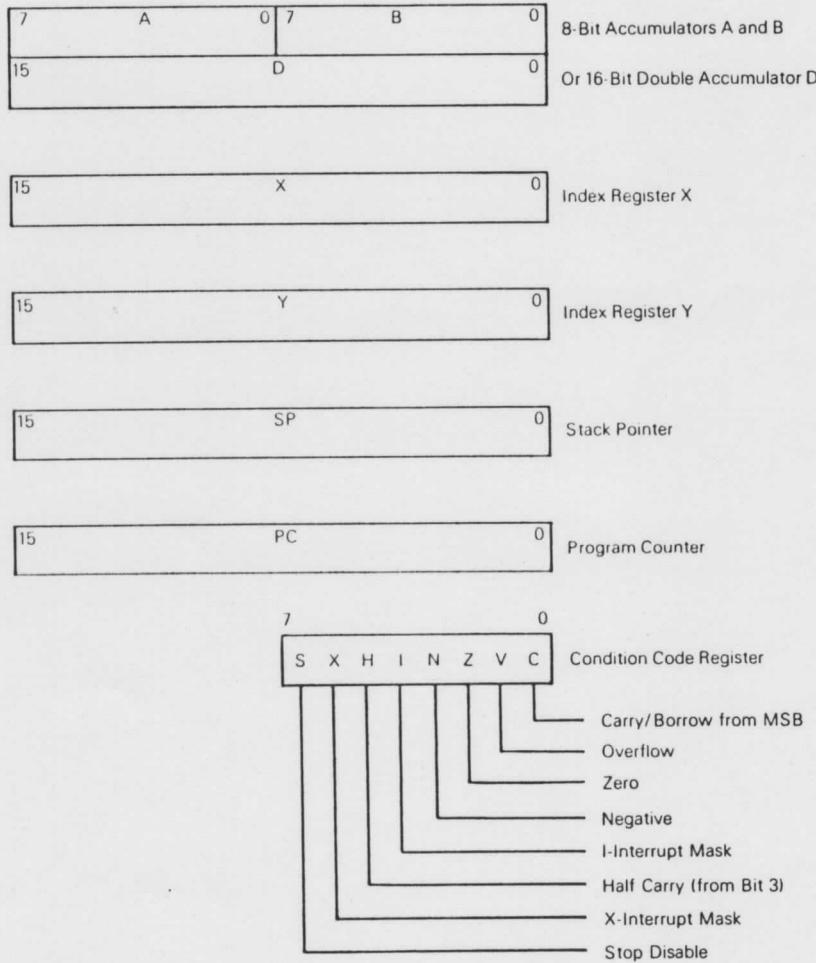
MC68HC11A4

memory, or on-chip peripheral configurations. Consequently, this CPU can be treated as an independent processor communicating with these internal subsystems when operating in the single-chip mode. However, when the MC68HC11A4 is operating in the extended multiplexed mode, it is capable of addressing external memory and

peripherals in addition to communicating with the on-chip subsystems.

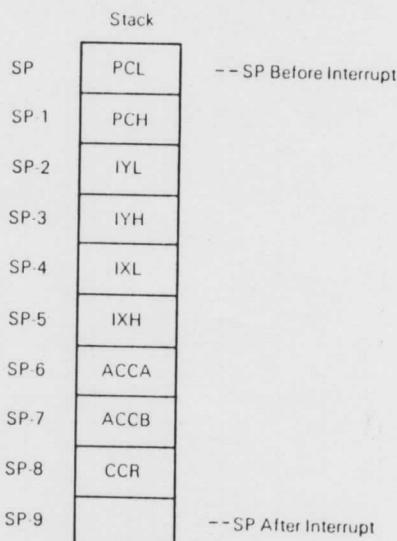
The MC68HC11A4 CPU has seven registers available to the programmers as shown in Figure 3. The interrupt stacking order is shown in Figure 4. The seven registers are discussed below.

FIGURE 3 — MC68HC11A4 PROGRAMMING MODEL



MC68HC11A4

FIGURE 4 – INTERRUPT STACKING ORDER



ACCUMULATOR A AND B

Accumulator A and accumulator B are general purpose 8-bit registers used to hold operands and results of arithmetic calculations or data manipulations. As in the MC6801, these two accumulators can be concatenated into a single double-byte accumulator called the D accumulator.

INDEX REGISTER X (IX)

The 16 bit IX register is used during indexed modes of addressing. It provides a 16 bit indexing value which may be added to an 8 bit offset provided in an instruction to create an effective address. The IX register can also be used as a counter or as a temporary storage area.

INDEX REGISTER Y (IY)

The 16-bit IY register is also used during indexed modes of addressing similar to the IX register; however, most instructions using the IY register require an extra byte of machine code and a cycle of execution time since they are two byte opcodes. The IY register can also be used as a counter or as a temporary storage in the same manner as the IX register.

STACK POINTER (SP)

The stack pointer (SP) is a 16-bit register that contains the address of the next free location on the stack. The stack is configured as a sequence of last-in-first-out read/write registers which allow important data to be stored during interrupts and subroutine calls. Each time a new byte is added to the stack (a push), the SP is decremented, whereas, each time a byte is removed from the stack (a pull) the SP is incremented. The address contained in the SP also indicates the location at which the accumulators (A and B), IX, and IY can be stored during certain instructions.

PROGRAM COUNTER (PC)

The program counter is a 16-bit register that contains the address of the next instruction to be executed.

CONDITION CODE REGISTER (CCR)

The condition code register is an 8-bit register in which each bit signifies the results of the instruction just executed. These bits can be individually tested by a program and a specific action can be taken as a result of the test. Each individual condition code register bit is explained below.

Carry/Borrow (C)

The C bit is set if there was a carry or borrow out of the arithmetic logic unit (ALU) during the last arithmetic operation. The C bit is also affected during shift and rotate instructions.

Overflow (V)

The overflow bit is set if there was an arithmetic overflow as a result of the operation; otherwise, the V bit is cleared.

Zero (Z)

The zero bit is set if the result of the last arithmetic, logic, or data manipulation was zero; otherwise, the Z bit is cleared.

Negative (N)

The negative bit is set if the result of the last arithmetic, logic, or data manipulation was negative (b7 of result equal to a logic one); otherwise, the N bit is cleared.

I Interrupt Mask (I)

The I interrupt mask bit is set either by hardware or program instruction to disable (mask) all maskable interrupt sources (both external and internal).

Half Carry (H)

The half carry bit is set to a logic one when a carry occurs between bits 3 and 4 of the arithmetic logic unit during an ADD, ABA, or ADC instruction; otherwise, the H bit is cleared.

X Interrupt Mask (X)

The X interrupt mask bit is set only by hardware (Reset or \overline{XIRQ}); and it is cleared only by program instruction (TAP or RTI).

Stop Disable (S)

The stop disable bit is set to disable the STOP instruction, and cleared to enable the STOP instruction. The S bit is program controlled. The STOP instruction is treated as no operation (NOP) if the S bit is set.

FUNCTIONAL PIN DESCRIPTION

The below pin descriptions do not include the I/O ports. They are discussed separately under INPUT/OUTPUT PORTS.

MC68HC11A4

V_{DD} AND V_{SS}

Power is supplied to the MC68HC11A4 using these two pins. V_{DD} is power input (+5 V) and V_{SS} is the power return path.

RESET

This active low bi-directional control pin is used as an input to initialize the MC68HC11A4 to a known start-up state, and as an open-drain output to indicate an internal failure has been detected in either the clock monitor or computer operating properly (COP) circuit.

XTAL, EXTAL

These two inputs provide for an interface with either a crystal input or a CMOS compatible clock to control the MC68HC11A4 internal clock generator circuitry. The frequency applied to these pins should be four times the desired internal clock rate since an internal divide-by-four circuit determines the actual E-clock rate. When a crystal is used, a 25 picofarad capacitor should be connected between V_{SS} and each of these two pins (XTAL and EXTAL); however, if a CMOS compatible external clock is used, the signal should be connected to the EXTAL pin and the XTAL pin should be left disconnected.

E

The E pin provides an output for the internally generated E clock which can be used as a timing reference. The frequency of the E output is actually one fourth that of the input frequency at the XTAL and EXTAL pins. In general when the E pin is low, an internal process is taking place and, when high, data is being accessed. This output becomes inactive during the power-saving wait mode if the MC68HC11A4 is operating in the single chip or bootstrap modes (see MODA/LIR, MODB/VPP description below).

IRQ

The IRQ pin provides a means for requesting asynchronous interrupts to the MC68HC11A4. The IRQ interrupt input is program selectable with a choice of either negative edge sensitive or level-sensitive triggering. The IRQ interrupt input is always configured to level-sensitive triggering during reset. The IRQ pin requires an external resistor to V_{DD}. The MCU completes the current instruction before responding to an interrupt request on the IRQ pin.

If IRQ is low and the interrupt mask bit (I bit) in the condition code register is clear, the MCU begins an interrupt sequence at the end of the current instruction.

XIRQ

The XIRQ pin provides a means for requesting asynchronous non-maskable interrupts to the MC68HC11A4, after a power-on reset. During reset (including power-on reset), the X bit in the condition code register is set and the XIRQ interrupt is masked to preclude interrupts on this line until MCU operation is stabilized. The X bit may then be cleared by program control (using the transfer accumulator A instruction, TAP); however, the X bit can only be set again by reset or by recognition of a hardware XIRQ interrupt. Once the X bit in the condition code register is cleared, an interrupt on the XIRQ pin will be serviced as soon as the MCU

completes the current instruction. When an XIRQ interrupt is recognized, on-chip hardware automatically sets the X bit. The X bit can be cleared either as part of interrupt routine by the TAP instruction (nested interrupt) or by the return from interrupt instruction. The XIRQ pin requires an external resistor to V_{DD}.

The XIRQ input may also be used to return the MCU to normal operation from the low-power stop mode by applying a low level to the XIRQ pin. If the X bit in the condition code register is cleared and the MCU is in the stop mode, a low input on the XIRQ brings the MCU out of the stop mode and operation resumes with the stacking operation leading to service of the XIRQ request. If the X bit is set and the MCU is in the stop mode, a low input on the XIRQ brings the MCU out of the stop mode and operation resumes with the program instruction following the STOP instruction.

MODA/LIR, MODB/VPP

These pins have alternate functions, MODA and MODB controlling one function, V_{PP} controlling an alternate function, and LIR used for an alternate function.

MODA, MODB

During reset these two pins are used to control the two basic operating modes of the MC68HC11A4 plus two special operating modes. The modes versus MODA and MODB inputs are shown in the table below.

MODB	MODA	Mode Selected
V _{DD}	0	Single Chip (Mode 0)
V _{DD}	1	Expanded Multiplexed (Mode 1)
#	0	Special Bootstrap
#	1	Special Test

1 = Logic High

0 = Logic Low

= 1.4 Times V_{DD} or Higher

VPP

In addition to the MODA function, the MODA/VPP pin is also used to supply the programming voltage for programming the internal EEPROM. Changing the voltage applied to this pin after reset has no affect on mode selection.

LIR

In addition to the MODA function, the MODA/LIR pin provides an output as an aid in debugging once reset is completed. The LIR pin goes to an active low during the first E-clock cycle of each instruction and remains low for the duration of that cycle (opcode fetch). Some MC68HC11A4 opcodes are two consecutive bytes long including a page 2 (PG2), page 3 (PG3), or page 4 (PG4) prebyte. For these instructions LIR goes low for only the first (prebyte) opcode byte fetch.

NOTE

The LIR output will not go low for at least two E-clock cycles after reset because of the reset vector fetch.

MC68HC11A4

V_{REFL}, V_{REFH}

These two pins provide the reference voltage for the analog-to-digital converter.

R/W/STRB

This pin provides two different functions depending on the operating mode. In single-chip mode the pin provides the STRB (output strobe) function and in the expanded multiplexed mode it provides the R/W (read/write) function.

In the single-chip mode the STRB pin acts as a programmable strobe. This strobe can also be used to provide a data acknowledge (handshake) to a parallel I/O device.

In the expanded multiplexed mode the R/W (read/write) is used to control the direction of transfers on the external data bus. A low level (write) on the R/W pin enables the data bus output drivers to the external data bus. A high level (read) on this pin forces the output drivers to a high-impedance state and data is read from the external bus.

AS/STRA

This pin provides two different functions depending on the operating mode. In single-chip mode, the pin provides the STRA (input strobe) function and in the expanded multiplexed mode it provides the AS (address strobe) function.

In the single-chip mode, the STRA pin acts as a programmable input strobe. This input is also used with STRB and port C for full handshake modes of parallel I/O.

In the expanded multiplexed mode the AS (address strobe) output may be used to demultiplex the address and data signals at port C.

INPUT/OUTPUT PORTS

There are five 8 bit ports on the MC68HC11A4 MCU. All of these ports serve more than one purpose depending on the mode configuration of the MCU. A summary of the pins versus function and mode is provided in Table 1 and discussed in the following paragraphs. Because the port functions are controlled by the particular mode selected, each port is discussed for its function(s) during the mode of operation.

SINGLE-CHIP MODE

In the single-chip mode the MC68HC11A4 functions as a monolithic microcomputer without external address or data buses. In this mode, four of these ports (A, B, C, D) are configured as parallel I/O data ports. Port E can be used for general purpose static inputs and/or analog-to-digital converter channel inputs.

Port A

In all operating modes (including the single-chip mode) port A may be configured for: three input capture functions (IC1, IC2, IC3), four output compare functions (OC2, OC3, OC4, OC5), and a pulse accumulator input (PA1) or a fifth output compare function (OC1).

Each of the input capture pins provide for a transitional input which is used to latch a timer value into a 16-bit read-only register (input capture register). The value latched by an input capture corresponds to the value of the free running counter which is part of the timer system. External devices provide the transitional inputs and internal decoders determine which input transition edge (rising, falling, or either) is sensed.

Each of the output compare pins provide for an output whenever a match is made between the value in the free-running counter (in the timer system) and a value loaded into the particular 16-bit output compare register. The outputs can be used externally to indicate that a certain period of time has elapsed.

When port A pin 7 (PA7) is configured as a pulse accumulator input (PA1), the external input pulses are applied to a pulse accumulator register within the MC68HC11A4.

Each port A pin that is not used for its alternate timer function, as described above, may be used as a general purpose input or output line.

Port B

In the single-chip mode, all of the port B pins are general purpose output pins. During MCU read cycles the levels sensed at the input side of the port B output drivers is read. Port B may also be used in a simple strobed output mode where the STRB (port D bit 7) pulses each time port B is written.

Port C

In the single-chip mode, all port C pins are general purpose input/output pins. Port C inputs can be latched by the STRA input (at port D bit 6). Port C may also be used in full handshake modes of parallel I/O where the STRA input and STRB output act as handshake control lines.

Port D

In the single-chip mode port D bits 0-5 may be used for general I/O or with the serial communications interface (SCI) and serial peripheral interface (SPI) subsystems. Bits 6 and 7 are used as handshake control signals for ports B and C.

Bit 0 is the receive data input (RxD) for the serial communication interface (SCI).

Bit 1 is the transmit data output (TxD) for the SCI. Bits 2 through 5 are dedicated to the serial peripheral interface (SPI). Bit 2 is the master-in-slave-out (MISO) line; this pin is an input when the SPI is configured as a master device and an output when configured as a slave device. Bit 3 is the master-out-slave-in (MOSI) line; this pin is an output when the SPI is configured as a master device and an input when configured as a slave device. Bit 4 is the serial clock (SCK) and is an output when the SPI is configured as a master and an input when configured as a slave device. Bit 5 is the slave select (SS) input which receives an active low signal to enable a slave device to accept SPI data.

Bit 6 (STRA) and 7 (STRB) are discussed in **FUNCTIONAL PIN DESCRIPTION**.

MC68HC11A4

TABLE 1 — PORT SIGNAL SUMMARY

Port-Bit	Single-Chip Modes 0 and Bootstrap Mode	Expanded Multiplexed Mode 1 and Special Test Mode
A-0	PA0/IC3	PA0/IC3
A-1	PA1/IC2	PA1/IC2
A-2	PA2/IC1	PA2/IC1
A-3	PA3/OC5/and-or OC1	PA3/OC5/and-or OC1
A-4	PA4/OC4/and-or OC1	PA4/OC4/and-or OC1
A-5	PA5/OC3/and-or OC1	PA5/OC3/and-or OC1
A-6	PA6/OC2/and-or OC1	PA6/OC2/and-or OC1
A-7	PA7/PA1/and-or OC1	PA7/PA1/and-or OC1
B-0	PB0	A8
B-1	PB1	A9
B-2	PB2	A10
B-3	PB3	A11
B-4	PB4	A12
B-5	PB5	A13
B-6	PB6	A14
B-7	PB7	A15
C-0	PC0	A0/D0
C-1	PC1	A1/D1
C-2	PC2	A2/D2
C-3	PC3	A3/D3
C-4	PC4	A4/D4
C-5	PC5	A5/D5
C-6	PC6	A6/D6
C-7	PC7	A7/D7
D-0	PD0/RxD	PD0/RxD
D-1	PD1/TxD	PD1/TxD
D-2	PD2/MISO	PD2/MISO
D-3	PD3/MOSI	PD3/MOSI
D-4	PD4/SCK	PD4/SCK
D-5	PD5/SS	PD5/SS
D-6	STRA	AS
D-7	STRB	R/W
E-0	PE0/AN0	PE0/AN0
E-1	PE1/AN1	PE1/AN1
E-2	PE2/AN2	PE2/AN2
E-3	PE3/AN3	PE3/AN3
E-4	PE4/AN4 ##	PE4/AN4 ##
E-5	PE5/AN5 ##	PE5/AN5 ##
E-6	PE6/AN6 ##	PE6/AN6 ##
E-7	PE7/AN7 ##	PE7/AN7 ##

- not bonded in 48 pin variations

Port E

In all operating modes (including the single-chip mode), port E is used for general purpose static inputs and/or analog-to-digital (A/D) channel inputs. Port E should not be read as static inputs while an A/D conversion is actually taking place.

NOTE

On 48-pin packaged versions of the MC68HC11A4, the four most significant bits of port E are not connected to pins.

EXPANDED MULTIPLEXED MODE

In the expanded multiplexed mode, the MC68HC11A4 has the capability of accessing a 64K byte address space. The

total address space includes the same on-chip memory address as for single-chip mode plus external peripheral devices. In this mode ports B, C, and bits 6 and 7 of port D are configured as a memory expansion bus.

Port A

In all operating modes (including the expanded multiplexed mode), port A may be configured for: three input capture functions (IC1, IC2, IC3), four output compare functions (OC2, OC3, OC4, OC5), and a pulse accumulator input (PA1) or a fifth output compare function (OC1).

Each of the input capture pins provide for a transitional input which is used to latch a timer value into a 16-bit read-only register (input capture register). The value latched by an input capture corresponds to the value of a free running

MC68HC11A4

counter which is part of the timer system. External devices provide the transitional inputs and internal decoders determine which input transition edge (rising, falling, or either) is sensed.

Each of the output compare pins provide for an output whenever a match is made between the value in the free running counter (in the timer system) and a value loaded into the particular 16 bit output compare register. The outputs can be used externally to indicate that a certain period of time has elapsed.

When port A pin 7 (PA7) is configured as a pulse accumulator input (PAII), the external input pulses are applied to a pulse accumulator register within the MC68HC11A4.

Each port A pin that is not used for its alternate timer function as described above, may be used as a general purpose input or output line.

Port B

In the expanded multiplexed mode, all of the port B pins act as high order address output pins. During each MCU cycle, bits 8 through 15 of the address are output on the PB0-PB7 lines respectively.

Port C

In the expanded multiplexed mode, all port C pins are configured as multiplexed address/data pins. During the address portion of each MCU cycle, bits 0 through 7 of the address are output on the PC0-PC7 lines. During the data portion of each MCU cycle (E high), bits 0 through 7 (D0-D7) are bidirectional data pins controlled by the R/W signal.

Port D

In the expanded multiplexed mode port D bits 0-5 may be used for general I/O or with the serial communications interface (SCI) and serial peripheral interface (SPI) subsystems. Bits 6 and 7 act as expansion bus control lines AS and R/W respectively.

Bit 0 is the receive data input (RxD) for the serial communications interface (SCI).

Bit 1 is the transmit data output (TxD) for the SCI.

Bits 2 through 5 are dedicated to the serial peripheral interface (SPI). Bit 2 is the master-in-slave-out (MISO) line; this pin is an input when the SPI is configured as a master device and an output when configured as a slave device. Bit 3 is the master-out-slave-in (MOSI) line; this pin is an output when the SPI is configured as a master device and an input when configured as a slave device. Bit 4 is the serial clock (SCK) and is an output when the SPI is configured as a master and an input when configured as a slave device. Bit 5 is the slave select (\overline{SS}) input which receives an active low signal to enable a slave device to accept SPI data.

Bit 6 (AS) and 7 (R/W) are discussed in **FUNCTIONAL PIN DESCRIPTION**.

Port E

In all operating modes (including the expanded multiplexed mode), port E is used for general purpose static inputs and/or analog-to-digital (A/D) channel inputs. Port E should not be read as static inputs while an A/D conversion is actually taking place.

NOTE

On 48-pin packaged versions of the MC68HC11A4, the four most significant bits of port E are not connected to external pins.

BOOTSTRAP MODE

In the bootstrap mode all I/O port pins function the same as in the single-chip mode. Operational differences are discussed in **OPERATING MODES**.

TEST MODE

In the test mode all I/O port pins function the same as in the expanded multiplexed mode. Operational differences are discussed in **OPERATING MODES**.

INTERRUPTS

The MC68HC11A4 MCU interrupts can be generated by any of four different basic methods: (1) by presenting the appropriate external signal, (2) by enabling interrupts from the programmable timer output compare or input capture, serial communication interface, serial peripheral interface timer overflow, pulse accumulator, or parallel I/O; (3) by executing a software interrupt (SWI) instruction; or (4) by detection of an illegal opcode.

The program may also be interrupted by: (1) detection of a timeout in the computer operating properly (COP) circuit, (2) clock monitor detects loss of the E-clock or a low frequency E-clock, or (3) by a reset. The above three methods of interrupting the program result in fetching a reset vector rather than an interrupt vector; however, they do interrupt the program.

When an external or internal (hardware) interrupt occurs, the interrupt is not serviced until the current instruction being executed is completed. Until the current instruction is complete, the interrupt is considered pending. After completion of current instruction execution, unmasked interrupts may be serviced in accordance with an established fixed hardware priority circuit; however, one I bit related interrupt source may be elevated to the highest I bit priority position in the circuit.

Seventeen hardware interrupts and one software interrupt (excluding reset type interrupts) can be generated from all of the possible sources. The interrupts can be divided into two basic categories, maskable and non-maskable. In the MC68HC11A4 fifteen of the interrupts can be masked using the condition code register I bit. In addition to being maskable by the I bit in the condition code register, all of the on-chip interrupt sources are individually maskable by control bits. The software interrupt (SWI instruction) is a non-maskable instruction rather than a maskable interrupt source. The last interrupt (external input to the XIRQ pin) is considered as a non-maskable interrupt because once enabled, it cannot be masked by software; however, it is masked during reset and upon receipt of an interrupt at the XIRQ pin. Table 2 provides a list of each interrupt, its vector location in ROM, and the actual condition code register bit that masks it. A discussion of the various interrupts is provided below.

MC68HC11A4

TABLE 2 – INTERRUPT VECTOR ASSIGNMENTS

Vector Address	Interrupt Source	Masked By
FFC0, C1 	Reserved 	—
FFD4, D5 FFD6, D7	Reserved SCI Serial System	1 Bit
FFD8, D9 FFDA, DB FFDC, DD FFDE, DF	SPI Serial Transfer Complete Pulse Accumulator Input Edge Pulse Accumulator Overflow Timer Overflow	1 Bit 1 Bit 1 Bit 1 Bit
FFE0, E1 FFE2, E3 FFE4, E5 FFE6, E7	Timer Output Compare 5 Timer Output Compare 4 Timer Output Compare 3 Timer Output Compare 2	1 Bit 1 Bit 1 Bit 1 Bit
FFE8, E9 FFEA, EB FFEC, ED FFEE, EF	Timer Output Compare 1 Timer Input Capture 3 Timer Input Capture 2 Timer Input Capture 1	1 Bit 1 Bit 1 Bit 1 Bit
FFF0, F1 FFF2, F3 FFF4, F5 FFF6, F7	Real Time Interrupt IRQ (External Pin or Parallel I/O) XIRQ Pin (Pseudo Non-maskable Interrupt) SWI	1 Bit 1 Bit X Bit * None
FFF8, F9 FFFA, FB FFFC, FD FFFE, FF	Illegal Op-Code Trap COP Failure (Reset) COP Clock Monitor Fail (Reset) RESET	None None None None

TIMER INTERRUPTS

The timer system provides nine of the fifteen interrupt possibilities: five output compare interrupts, three input capture interrupts, and a timer overflow interrupt.

The timer contains five 16-bit output compare registers which are program controlled and may be loaded with a number between \$0000-\$FFFF. The value in each output compare register is then compared to a 16-bit comparator, which is loaded from the timer free running counter, during each clock cycle. If a match is found between the 16-bit comparator value and the output compare register value, the corresponding output compare flag is set. When the output compare flag is set, a corresponding output compare interrupt may be generated and/or an external output may be generated at the corresponding port A pin(s). Port A outputs PA3 through PA7 are used as output pins for output compare functions OC1 through OC5.

In addition to the five output compare interrupts, the timer also provides for three input capture interrupts. The timer contains three 16-bit latch registers which are used to latch the value of the free running counter (in the timer) when an input capture edge is applied to the corresponding PA0-PA2 pin. The value of the free running counter is latched into the corresponding input capture register and an internal interrupt may be generated. The interrupt routine can then read the storage register and determine the time at which the input capture was detected.

The timer may also provide an interrupt when the free running counter changes value from \$FFFF to \$0000 (overflow).

The 16-bit free running counter repeats this change once for every 65,536 inputs from a prescaler circuit. The prescaler is programmable for either divide-by-1, divide-by-4, divide-by 8, or divide-by-16 of the MCU E-clock. Thus, the prescaler extends the actual range of the free running counter and the time between timer overflow interrupts from 216 to 2256 E-clock inputs to the prescaler.

REAL TIME INTERRUPTS

The real time interrupt is a maskable interrupt that occurs periodically at a rate of E/213, E/214, E/215, or E/216.

EXTERNAL INTERRUPTS

Two external interrupts are accessible using the IRQ and the XIRQ pins. The IRQ interrupt is a maskable interrupt while the XIRQ interrupt is considered a non-maskable interrupt; however, the XIRQ interrupt is masked during reset and immediately following receipt of an XIRQ interrupt signal. These interrupts are controlled by the I and X bits in the condition code register as discussed in **CENTRAL PROCESSING UNIT**.

SOFTWARE INTERRUPT (SWI)

The software interrupt is executed the same as any other instruction and will take precedence over interrupts only if the other interrupts are masked (I and X bits in the condition code register set). The SWI instruction is executed similar to other maskable interrupts in that it sets the I bit, CPU registers are stacked, etc.

MC68HC11A4

NOTE

The SWI instruction cannot be fetched as long as another interrupt is pending execution. However, once it is fetched no other interrupt can be honored until the first instruction in the SWI service routine is completed.

SERIAL PERIPHERAL INTERFACE (SPI) INTERRUPT

A serial peripheral interface (SPI) interrupt is generated when a serial data transfer between the MC68HC11A4 and an external device has been completed. This interrupt is masked if the condition code register I bit is set.

SERIAL COMMUNICATIONS INTERFACE (SCI) INTERRUPT

A serial communications interface (SCI) interrupt is generated if any one of the following occurs in the SCI:

1. Transmit data register is empty
2. Transmission of data is complete
3. Receive data register is full or an overflow occurred in the receive data register
4. Idle line detected by receiver.

The SCI interrupt is masked if the condition code register I bit is set.

PULSE ACCUMULATOR INTERRUPT

The pulse accumulator contains an 8-bit counter which is program controlled to either count input pulses (event counting) at PA7 or to count internal E/64 clocks subject to an enable signal at PA7 (gated time accumulation). When the counter has an overflow from \$FF to \$00 a pulse accumulator overflow interrupt is generated provided the I bit in the condition code register is clear.

When the input to the pulse accumulator is a gate input at PA7 for counting internal E/64 clocks, the trailing edge of the gate signal (end of counting cycle) can generate an interrupt. This pulse accumulator input edge interrupt is generated provided the I bit in the condition code register is clear. Refer to PULSE ACCUMULATOR for more information.

PARALLEL I/O INTERRUPT

The parallel I/O subsystem can generate an interrupt which uses the same vector as the IRQ interrupt. The purpose of sharing the IRQ vector is to allow external emulation of the parallel I/O subsystem in expanded multiplexed modes.

RESETS

The MC68HC11A4 MCU has four possible types of reset: an active low external reset pin (RESET), a power-on reset function, a computer operating properly (COP) watchdog timer reset, and a clock monitor reset.

RESET PIN

The RESET pin is used to reset the MCU to provide an orderly software startup procedure. To request an external reset, the RESET pin must be held low for eight Ecyc (two Ecyc if internal resets are not used).

POWER-ON RESET

The power-on reset occurs when a positive transition is detected on VDD. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drops in power supply voltage. There is no provision for power-down reset. If the external RESET pin is low at the end of the power-on delay time, the processor remains in the reset condition until RESET goes high.

COMPUTER OPERATING PROPERLY (COP) RESET

The MC68HC11A4 MCU contains a watchdog timer which will time itself out if not reset within a specific time by a program reset sequence. If for any reason the COP watchdog timer is allowed to timeout, it generates an MCU reset which is functionally similar to pulling the RESET pin low.

A control bit, which is implemented in an EEPROM cell of the system configuration register, is used to enable (or disable) the COP reset function. When this bit is clear, the COP reset function is disabled; if set, the COP reset is enabled.

CLOCK MONITOR RESET

The MC68HC11A4 MCU contains a clock monitor circuit which measures the E-clock input frequency. If the E clock input rate is high enough, then the clock monitor does not time out. However, if the E clock signal is lost, or its frequency falls below 200 kHz, then an MCU reset is generated which is functionally similar to pulling the RESET pin low.

A read-write control bit, which is implemented in the system configuration options register, is used to enable (or disable) the clock monitor reset. When this bit is clear, the clock monitor reset function is disabled; when set, the clock monitor reset is enabled.

STOP AND WAIT

The MC68HC11A4 MCU contains two programmable low-power operating modes; stop and wait. In the wait mode, the on-chip oscillator remains active together with other functions discussed below. In the stop mode, all clocks including the crystal oscillator are stopped.

WAI (WAIT) INSTRUCTION

The WAI instruction places the MC68HC11A4 MCU in a low power consumption (wait) mode. In the wait mode, the internal clock remains active, and the MCU enters one of four different variations of the wait mode. These variations, which depend upon the I bit in the condition register and whether or not the COP circuit is required in the system, include: (1) only the CPU turned off; (2) CPU and the E clock output buffer turned off; (3) CPU and timer system turned off; or (4) CPU, E output, and timer system all off.

During the wait mode, the CPU registers are stacked and processing is suspended until a qualified interrupt is detected. The actual qualified interrupt type is dependent upon which of the wait mode variations is selected. The qualified interrupt(s) required to bring the MCU out of the wait mode for each of the wait mode variations is shown below. In all cases, reset brings the MCU out of the wait mode; however, as in all resets, the system is reset and the start of MCU operation is determined by the reset vector.

MC68HC11A4

Wait Mode Variation	Qualified Interrupt
Only CPU Turned Off	IRQ, XIRQ, Any Internal Interrupt
CPU and E Clock Output Buffers Turned Off	IRQ, XIRQ, Any Internal Interrupt
CPU and Timer System Turned Off	IRQ, XIRQ
CPU, E Clock Output Buffers, and Timer System Turned Off	IRQ, XIRQ

STOP INSTRUCTION

The STOP instruction places the MC68HC11A4 MCU in its lowest power consumption mode provided the S bit in the condition code register is clear. In the stop mode all clocks including the internal oscillator are stopped, causing all internal processing to be halted. To exit the stop mode and resume normal processing, a low level must be applied to one of the external interrupt pins (IRQ or XIRQ) or to the RESET pin. If an external interrupt is used at the IRQ input, it is only effective if the I bit in the condition code register is clear. If an external interrupt is applied at the XIRQ input, the MCU exits from the stop mode regardless of the state of the X bit in condition code register; however, the actual recovery sequence differs depending on the X bit. If the X bit is clear, the MCU starts up with the stacking sequence leading to normal service of the XIRQ request. If the X bit is set, then processing will continue with the instruction immediately following the STOP instruction and no XIRQ interrupt service routine is requested. As in the wait mode, a low input to the RESET pin will always result in an exit from the stop mode and the start of MCU operation is determined by the reset vector.

Since the oscillator is stopped in the stop mode, a restart delay may be required to allow for oscillator stabilization when exiting from the stop mode. If the internal oscillator is being used, this delay is required; however, if a stable external oscillator is being used, a control bit within the MCU may be used (cleared) to bypass the delay. If the delay bypass control bit is clear then the RESET pin would not normally be used for exiting the stop mode. In this case, the reset sequence sets the delay control bit and the restart delay will be imposed.

PROGRAMMABLE TIMER SYSTEM

The timer system in the MC68HC11A4 uses a "time-of-day" approach in that all timing functions are related to a single 16-bit free running counter. The free running counter is clocked by the output of a programmable prescaler (divide-by-1, 4, 8, or 16) which is in turn clocked by the MCU E clock. Functions available within the MC68HC11A4 timer include: three input capture functions and five output compare functions.

The capabilities of the programmable timer are obtained using the following registers:

1. Prescaler (divide-by-1, 4, 8, or 16)
2. Free Running Counter (16-bit)

3. Input Capture (three 16-bit registers)
4. Output Compare (five 16-bit registers)
5. Main Timer Control and Status Registers

PRESCALER AND FREE RUNNING COUNTER

The key element in the timer system is a 16-bit free running counter with its associated programmable prescaler (divide-by-1, 4, 8, or 16). The free running counter is clocked by the output of the prescaler which is in turn clocked by the E clock. The free running counter can be read by software at any time without affecting its value since it is clocked and read on opposite half cycles of the MPU E clock. The free running counter is cleared to \$0000 during reset and is a read-only register (except in the test or bootstrap mode where this feature is used in factory testing).

The 16-bit free running counter repeats every 65,536 counts (prescaler output) and when the count changes from \$FFFF to \$0000 a timer overflow flag bit is set. Setting the timer overflow flag bit also generates an internal interrupt if the overflow interrupt enable bit is set.

Input Capture Functions

There are three separate 16-bit read-only input capture registers which are not affected by reset. Each of these registers is used to latch the value of the free running counter when a selected transition at an external pin is detected. External devices provide the inputs on the PA0-PA2 pins, and an interrupt can be generated when an input capture edge is detected. The time of detection can be read from the appropriate register as part of the interrupt routine.

Output Compare Functions

There are five separate 16-bit read/write output compare registers which are initialized to \$FFFF at reset. The value written into the output compare register is compared to the free running counter value during each MCU E clock cycle. If a match is found between the two values, the particular output compare flag bit is set and an interrupt is generated provided that particular interrupt is enabled.

In addition to the interrupt, a specified action may be initiated at a timer output pin(s). For OC1, the output action to be taken, when a match is found, is controlled by a 5-bit mask register and a 5-bit data register. The 5-bit mask register specifies which timer port outputs are to be affected and the 5-bit data register specifies the data to be placed on the affected output pins. For OC2 through OC5, one specific timer output is affected as controlled by four 2-bit fields in a timer control register. Specific actions include: (1) timer disconnect from output pin logic, (2) toggle output compare line, (3) clear output compare line to zero, or (4) set output compare line to one.

PULSE ACCUMULATOR

The pulse accumulator is an 8-bit counter that can operate in either of two modes depending on the state of a control bit. These include the event counting mode or the gated time accumulation mode.

The pulse accumulator control register contains four bits which enable and configure the pulse accumulator system. One bit enables the counter. One bit determines whether the

MC68HC11A4

PA7/PA1 pin will be an input or an output. A third bit specifies the event counting mode or the gated time accumulation mode, and the fourth bit determines which edge of the PA1 input is the active one. The 8 bit counter counts from \$00 to \$FF and when it overflows from \$FF to \$00 a flag bit is set. This results in a hardware interrupt provided the pulse accumulator overflow interrupt enable bit is set.

In the event counting mode, the 8-bit counter is clocked to increasing values by an external (PA1) pin input (PA7). In the gated time accumulation mode, the 8-bit counter is clocked to increasing values by the MCU E clock (divided by 64) provided the proper gating signal is applied to an external (PA1) pin input (PA7).

SERIAL COMMUNICATIONS INTERFACE (SCI)

The serial communications interface (SCI) allows the MC68HC11A4 to be efficiently interfaced with peripheral devices that require an asynchronous serial data format. The SCI in the MC68HC11A4 is provided with a standard NRZ format with a variety of baud rates. The baud rate is derived from the crystal clock circuit and interface with peripheral devices is accomplished using port D pins PD0 for receive data (RxD) and PD1 for transmit data (TxD).

BAUD RATE GENERATION

The actual baud rate generation circuit contains a programmable prescaler and divider which is clocked by the MCU E clock. A programmable baud rate register is used to provide the various divide ratios used in the baud rate generator prescaler and divider. This scheme of baud rate generation allows for selection of many different standard baud rates, all of which are controlled by the crystal oscillator.

DATA FORMAT

Receive data (RxD) in or transmit data (TxD) out is the serial data which is presented between the input pin (PD0) and the internal data bus, and between the internal data bus and the output pin (PD1). The data format requires:

1. An idle line which is in the high state (logic one) prior to transmission/reception of a message.
2. A start bit (logic zero) which is transmitted/received indicating the start of a message.
3. Data is transmitted and received least-significant bit first.
4. A stop bit (logic one in the tenth or eleventh bit position) indicates the byte is complete.
5. A break is defined as the transmission or reception of a logic zero for some multiple of the data format.

The data format word length may consist of either ten or eleven bits. Selection of the word length is controlled by a single bit in a control register within the SCI. If this control bit is clear, the data contains a start bit, eight data bits, and a stop bit. If this control bit is set, there is a start bit, nine data bits, and a stop bit.

TRANSMIT OPERATION

The SCI transmitter includes a parallel data register and a serial shift register. This is referred to as a double buffered system in that besides the character being shifted out serially, another character is already waiting to be loaded into the serial shift register. The output of the transmit serial shift register is applied to the TxD output pin (PD1) as long as a transmit enable bit is set.

RECEIVE OPERATION

Receive data in (RxD) is serial data which is presented to the input pin (PD0). An advanced data recovery scheme is used to distinguish valid data from noise in the serial data stream. In this manner the data input can be selectively sampled to detect receive data and then verify that the data is valid. Data is received in a serial shift register and is transferred to a parallel register as a complete byte. This is referred to as a double buffered system in that besides the character already in the parallel register, another is being shifted in serially.

WAKE-UP FEATURE

The wake-up feature allows a receiver(s) to "sleep" until a specific action takes place. In a typical multiprocessor configuration, the software protocol will usually identify the addressee(s) at the beginning of a message. This wake-up feature allows uninterested MPUs to ignore incoming messages. The MC68HC11A4 SCI permits this wake-up feature by either of two methods: idle line wake-up or address mark wake-up.

In idle line wake-up, all receivers wake up whenever an idle line is detected; however, if a receiver does not recognize its address in the first frame of a message it may ignore the rest of the message by invoking the wake-up feature. In this wake up method, transmitter software must provide for the required idle string between consecutive messages and prevent it from occurring within messages.

In the address mark wake-up, all serial frames consist of seven (or eight) information bits plus a most-significant bit (MSB) which is used to indicate an address frame if the MSB is a logic one. The first frame of each message is an address frame which wakes up all receivers in the system. All receivers evaluate this marked address frame to determine which receiver(s) the message is intended for. If a receiver determines that a message is not intended for it, it invokes the receiver wake-up function so that no additional program overhead is required for the rest of the message.

INTERRUPT FLAGS

The serial communications interface (SCI) generates a hardware interrupt (SCI interrupt) whenever any one of several flags is set and its corresponding interrupt enable bit is also set. These flags which are discussed below include:

1. Transmit Data register empty
2. Transmission complete
3. Idle line detected
4. Receive data register full or overrun error detected.

MC68HC11A4

The transmit data register empty (TDRE) bit is set to indicate that the transmit parallel data register contents have been transferred to the transmit serial shift register. If the corresponding interrupt enable bit (transmit interrupt enable) is set then an SCI interrupt is generated.

The transmission complete (TC) bit is set when the transmitter no longer has any meaningful information to transmit; i.e., no data in the serial shifter, no queued preamble, and no queued break. If the transmitter is enabled when TC is set, the serial line will go idle (continuous mark).

The idle line detected (IDLE) bit is set whenever a receiver detects a receiver idle line. This could indicate the end of a message, the preamble of a new message, or resynchronization with the transmitter. If the corresponding interrupt enable bit (idle line interrupt enable) is set then an SCI interrupt is generated.

The receiver data register full (RDRF) bit is set whenever the receiver serial shift register contents are transferred to the serial communications data register. If the corresponding interrupt enable bit (receive interrupt enable) is set then an SCI interrupt is generated.

The overrun error bit is set to indicate that the next byte is ready for transfer from the receive shift register to the receive data register but that register is already full (RDRF bit set). Data transfer is then inhibited until the OR (overrun) bit is cleared. As with the RDRF bit, an SCI interrupt is generated if the corresponding interrupt enable bit is set.

SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI) allows several MC68HC11A4 MCUs, or MC68HC11A4 MCUs plus peripheral devices, to be interconnected within a single "black box", on the same printed circuit board. In a serial peripheral interface, the MC68HC11A4 provides such features as:

- Full Duplex, Two, Three, or Four Wire Synchronous Transfers
- Master or Slave Operation
- Interface With Low Cost "Dumb" Peripherals
- Interface With Intelligent Peripherals on Master/Slave Basis
- Four Programmable Master Bit Rates
- Programmable Clock Polarity and Phase
- End of Transmission Interrupt Flag
- Write Collision Error Detection
- Master-Master Mode Fault Error Detection

Four basic signal lines are associated with the SPI system. These include a master-out slave-in (MOSI) line, a master-in slave-out (MISO) line, a serial clock (SCK) line, and a slave select (SS) line. Two master-slave system configurations are shown in Figure 5 and the basic signals (MOSI, MISO, SCK, and SS) are described below.

MASTER OUT SLAVE IN (MOSI)

The MOSI pin is configured as a data output in a master (mode) device and as a data input in a slave (mode) device.

In this manner data is transferred serially from a master to a slave on this line; most significant bit first, least significant last.

MASTER IN SLAVE OUT (MISO)

The MISO pin is configured as an input in a master (mode) device and as an output in a slave (mode) device. In this manner data is transferred serially from a slave to a master on this line; most significant bit first, least significant last.

SLAVE SELECT (SS)

The slave select (SS) is a fixed input which receives an active low signal that is generated by a master device to enable slave devices to accept data.

SERIAL CLOCK (SCK)

The serial clock is used to synchronize the movement of data both in and out of the device through its MOSI or MISO pins. The master and slave devices can exchange a byte of information during a sequence of eight clock pulses. The SCK is generated by the controlling master device and becomes an input on all slave devices to synchronize slave data transfer.

ANALOG-TO-DIGITAL (A/D) CONVERTER

The MC68HC11A4 contains an 8-channel, multiplexed input, successive approximation analog-to-digital converter with sample and hold. Two dedicated pins (VREFL, VREFH) are provided for the reference supply voltage input. These dedicated pins are used instead of the device power pins to increase accuracy of the A/D conversion.

The 8-bit A/D conversions of the MC68HC11A4 are accurate to within \pm one LSB ($\pm \frac{1}{2}$ LSB quantizing error and $\pm \frac{1}{2}$ LSB non-linearity error). Each conversion is accomplished in 50 MCU E clock cycles or less. An internal control bit allows selection of an internal conversion clock oscillator which allows the A/D to be used with very low MCU clock rates. A typical conversion cycle requires 25 to 50 microseconds to complete.

NOTE

In the 48-pin dual in-line package, four conversion channels are not implemented. These include channels four through seven.

ADDRESSING MODES

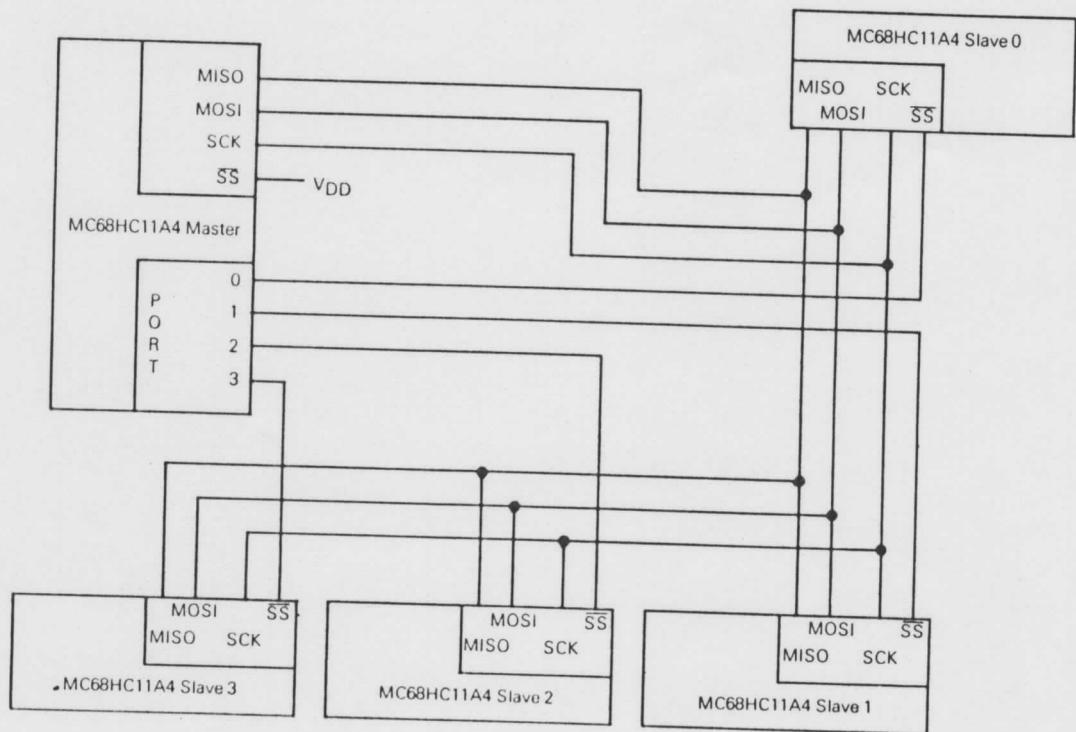
Six addressing modes can be used to reference memory; they include: immediate, direct, extended, indexed (with either of two 16-bit index registers and an 8-bit offset), inherent, and relative. Some instructions require an additional byte before the opcode to accommodate a multi-page opcode map; this byte is called a prebyte.

The following paragraphs provide a description of each addressing mode plus a discussion of the prebyte. In these descriptions the term effective address is used to indicate the address in memory from which the argument is fetched or stored, or from which execution is to proceed.

MC68HC11A4

FIGURE 5 - MASTER-SLAVE SYSTEM CONFIGURATION
(Sheet 1 of 2)

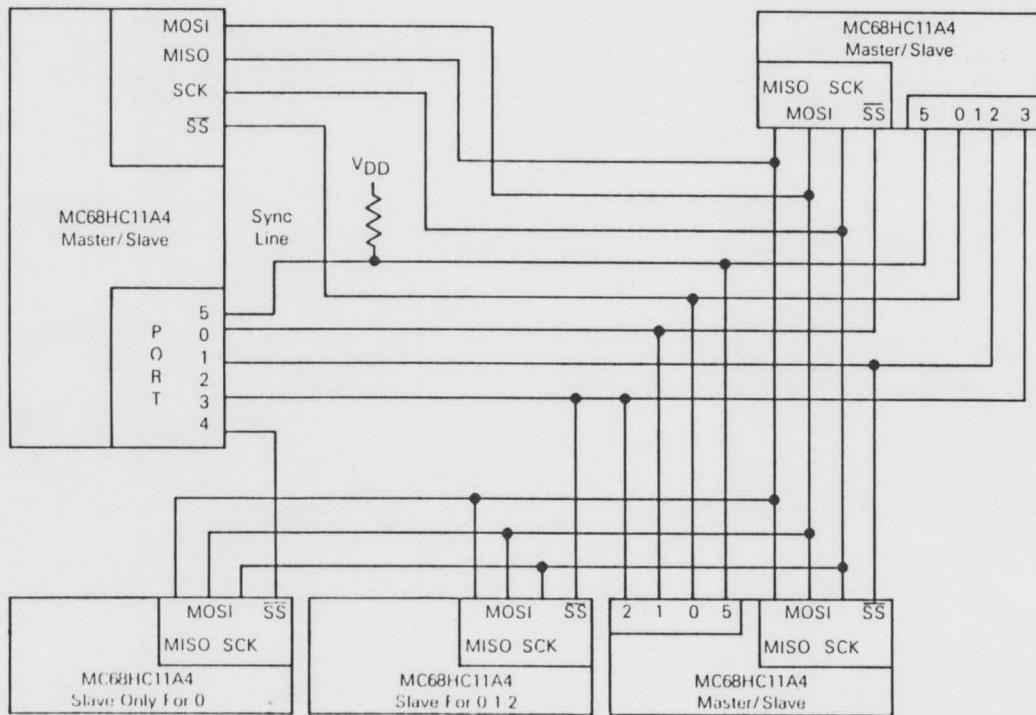
a. Single Master, Four Slaves



MC68HC11A4

FIGURE 5 - MASTER-SLAVE SYSTEM CONFIGURATION
(Sheet 2 of 2)

b. Three Master/Slave, Two Slaves



MC68HC11A4

IMMEDIATE ADDRESSING

In the immediate addressing mode, the actual argument is contained in the byte(s) immediately following the instruction where the number of bytes matches the size of the register. These are two, three, or four (if prebyte is required) byte instructions.

DIRECT ADDRESSING

In the direct addressing mode, the least significant byte of the operand address is contained in a single byte following the opcode and the most significant byte is assumed to be \$00. Direct addressing allows the user to access addresses \$0000 through \$00FF using two byte instructions and execution time is reduced by eliminating the additional memory access. In most applications, this 256 byte area is reserved for frequently referenced data. These are usually two or three (if prebyte is required) byte instructions.

EXTENDED ADDRESSING

In the extended addressing mode, the second and third bytes following the opcode contain the absolute address of the operand. These are three or four (if prebyte is required) byte instructions: one or two for the opcode and two for the effective address.

INDEXED ADDRESSING

In the indexed addressing mode, one of the index registers (X or Y) is used in calculating the effective address. In this case the effective address is variable and depends on two factors: (1) the current contents of the index register (X or Y) being used, and (2) the 8 bit unsigned offset contained in the instruction. This addressing mode allows referencing any memory location in the 64K byte address space. These are usually two or three (if prebyte is required) byte instructions, the opcode plus the 8 bit offset.

INHERENT ADDRESSING

In the inherent addressing mode, all of the information to execute the instruction is contained in the opcode. The

operands (if any) are registers and no memory reference is required. These are usually one or two byte instructions.

RELATIVE ADDRESSING

The relative addressing mode is used for branch instructions. If the branch condition is true and contents of the 8-bit signed byte following the opcode (the offset) is added to the contents of the program counter to form the effective branch address; otherwise, control proceeds to the next instruction. These are usually two byte instructions.

PREBYTE

In order to expand the number of instructions used in the MC68HC11A4, a prebyte instruction has been added to certain instructions. The instructions affected are usually associated with index register Y. The opcode instructions which do not require a prebyte could be considered as page 1 of the overall opcode map. The remaining opcodes could be considered as pages 2, 3, and 4 of the opcode map and would require a prebyte; \$18 for page 2, \$1A for page 3, and \$CD for page 4. Refer to **INSTRUCTION SUMMARY** for more detail.

INSTRUCTION SET

The central processing unit (CPU) in the MC68HC11A4 is basically a proper extension of the MC6801 CPU. In addition to its ability to execute all M6800 and M6801 instructions, the MC68HC11A4 CPU has a paged operation code (opcode) map with a total of 91 new opcodes. Major functional additions include a second 16-bit index register (Y register), two types of 16-by-16 divide instructions, a STOP instruction, and bit manipulation instructions.

Table 3 shows all MC68HC11A4 instructions in all possible addressing modes. For each instruction the operand construction is shown as well as the total number of machine code bytes and execution time in CPU E-clock cycles. Notes are provided at the end of Table 3 which explain the letters in the Operand and Execution Time columns of some instructions.

MC68HC11A4

TABLE 3 — MC68HC11A4 INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES

Source Form(s)	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Machine Code Bytes (Total)	Execution Time (Cycles)
		Opcode	Operand(s)		
ABA	INH	1B		1	2
ABX	INH	3A		1	3
ABY	INH	18 3A		2	4
ADCA (opr)	A IMM A DIR A EXT A IND, X A IND, Y	89 99 B9 A9 18	ii dd hh ii ff A9	2 2 3 2 3	2 3 4 4 5
ADCB (opr)	B IMM B DIR B EXT B IND, X B IND, Y	C9 D9 F9 E9 18	ii dd hh ii ff E9	2 2 3 2 3	2 3 4 4 5
ADDA (opr)	A IMM A DIR A EXT A IND, X A IND, Y	8B 9B BB AB 18	ii dd hh ii ff AB	2 2 3 2 3	2 3 4 4 5
ADDB (opr)	B IMM B DIR B EXT B IND, X B IND, Y	CB DB FB EB 18	ii dd hh ii ff EB	2 2 3 2 3	2 3 4 4 5
ADDD (opr)	IMM DIR EXT IND, X IND, Y	C3 D3 F3 E3 18	jj kk dd hh ii ff E3	3 2 3 2 3	4 5 6 6 7
ANDA (opr)	A IMM A DIR A EXT A IND, X A IND, Y	84 94 B4 A4 18	ii dd hh ii ff A4	2 2 3 2 3	2 3 4 4 5
ANDB (opr)	B IMM B DIR B EXT B IND, X B IND, Y	C4 D4 F4 E4 18	ii dd hh ii ff E4	2 2 3 2 3	2 3 4 4 5
ASL (opr)	EXT IND, X IND, Y	78 68 18	hh ii ff ff	3 2 3	6 6 7
ASLA	A INH	48		1	2
ASLB	B INH	58		1	2
ASLD	INH	05		1	3
ASR (opr)	EXT IND, X IND, Y	77 67 18	hh ii ff ff	3 2 3	6 6 7
ASRA	A INH	47		1	2
ASRB	B INH	57		1	2
BCC (rel)	REL	24	rr	2	3
BCLR (opr) (msk)	DIR IND, X IND, Y	15 1D 18	dd mm ff mm ff mm	3 3 4	6 7 8
BCS (rel)	REL	25	rr	2	3
BEQ (rel)	REL	27	rr	2	3
BGE (rel)	REL	2C	rr	2	3
BGT (rel)	REL	2E	rr	2	3

MC68HC11A4

TABLE 3 – MC68HC11A4 INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES (CONTINUED)

Source Form(s)	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Machine Code Bytes (Total)	Execution Time (Cycles)
		Opcode	Operand(s)		
BHI (rel)	REL	22	rr	2	3
BHS (rel)	REL	24	rr	2	3
BITA (opr)	A IMM	85	ii	2	2
	A DIR	95	dd	2	3
	A EXT	B5	hh II	3	4
	A IND, X	A5	ff	2	4
	A IND, Y	18	A5	3	5
BITB (opr)	B IMM	C5	ii	2	2
	B DIR	D5	dd	2	3
	B EXT	F5	hh II	3	4
	B IND, X	E5	ff	2	4
	B IND, Y	18	E5	3	5
BLE (rel)	REL	2F	rr	2	3
BLO (rel)	REL	25	rr	2	3
BLS (rel)	REL	23	rr	2	3
BLT (rel)	REL	2D	rr	2	3
BMI (rel)	REL	2B	rr	2	3
BNE (rel)	REL	26	rr	2	3
BPL (rel)	REL	2A	rr	2	3
BRA (rel)	REL	20	rr	2	3
BRN (rel)	REL	21	rr	2	3
BRCLR (opr) (msk) (rel)	DIR	13	dd mm rr	4	6
	IND, X	1F	ff mm rr	4	7
	IND, Y	18	1F	5	8
BRSET (opr) (msk) (rel)	DIR	12	dd mm rr	4	6
	IND, X	1E	ff mm rr	4	7
	IND, Y	18	1E	5	8
BSET (opr) (msk)	DIR	14	dd mm	3	6
	IND, X	1C	ff mm	3	7
	IND, Y	18	1C	4	8
BSR (rel)	REL	8D	rr	2	6
BVC (rel)	REL	28	rr	2	3
BVS (rel)	REL	29	rr	2	3
CBA	INH	11		1	2
CLC	INH	0C		1	2
CLI	INH	0E		1	2
CLR (opr)	EXT	7F	hh II	3	6
	IND, X	6F	ff	2	6
	IND, Y	18	6F	3	7
CLRA	A INH	4F		1	2
CLRB	B INH	5F		1	2
CLV	INH	0A		1	2
CMPA (opr)	A IMM	81	ii	2	2
	A DIR	91	dd	2	3
	A EXT	B1	hh II	3	4
	A IND, X	A1	ff	2	4
	A IND, Y	18	A1	3	5
CMPB (opr)	B IMM	C1	ii	2	2
	B DIR	D1	dd	2	3
	B EXT	F1	hh II	3	4
	B IND, X	E1	ff	2	4
	B IND, Y	18	E1	3	5

MC68HC11A4

TABLE 3 – MC68HC11A4 INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES (CONTINUED)

Source Form(s)	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Machine Code Bytes (Total)	Execution Time (Cycles)
		Opcode	Operand(s)		
COM (opr)	EXT	73	hh ll	3	6
	IND, X	63	ff	2	6
	IND, Y	18 63	ff	3	7
COMA	A INH	43		1	2
COMB	B INH	53		1	2
CPD (opr)	IMM	1A 83	jj kk	4	5
	DIR	1A 93	dd	3	6
	EXT	1A B3	hh ll	4	7
	IND, X	1A A3	ff	3	7
	IND, Y	CD A3	ff	3	7
CPX (opr)	IMM	8C	jj kk	3	4
	DIR	9C	dd	2	5
	EXT	BC	hh ll	3	6
	IND, X	AC	ff	2	6
	IND, Y	CD AC	ff	3	7
CPY (opr)	IMM	18 8C	jj kk	4	5
	DIR	18 9C	dd	3	6
	EXT	18 BC	hh ll	4	7
	IND, X	1A AC	ff	3	7
	IND, Y	18 AC	ff	3	7
DAA	INH	19		1	2
DEC (opr)	EXT	7A	hh ll	3	6
	IND, X	6A	ff	2	6
	IND, Y	18 6A	ff	3	7
DECA	A INH	4A		1	2
DEC B	B INH	5A		1	2
DES	INH	34		1	3
DEX	INH	09		1	3
DEY	INH	18 09		2	4
EORA (opr)	A IMM	88	ii	2	2
	A DIR	98	dd	2	3
	A EXT	B8	hh ll	3	4
	A IND, X	A8	ff	2	4
	A IND, Y	18 A8	ff	3	5
EORB (opr)	B IMM	C8	ii	2	2
	B DIR	D8	dd	2	3
	B EXT	F8	hh ll	3	4
	B IND, X	E8	ff	2	4
	B IND, Y	18 E8	ff	3	5
FDIV	INH	03		1	41
IDIV	INH	02		1	41
INC (opr)	EXT	7C	hh ll	3	6
	IND, X	6C	ff	2	6
	IND, Y	18 6C	ff	3	7
INCA	A INH	4C		1	2
INC B	B INH	5C		1	2
INS	INH	31		1	3
INX	INH	08		1	3
INY	INH	18 08		2	4
JMP (opr)	EXT	7E	hh ll	3	3
	IND, X	6E	ff	2	3
	IND, Y	18 6E	ff	3	4
JSR (opr)	DIR	9D	dd	2	5
	EXT	BD	hh ll	3	6
	IND, X	AD	ff	2	6
	IND, Y	18 AD	ff	3	7

MC68HC11A4

TABLE 3 — MC68HC11A4 INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES (CONTINUED)

Source Form(s)	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Machine Code Bytes (Total)	Execution Time (Cycles)
		Opcode	Operand(s)		
LDAA (opr)	A IMM	86	ii	2	2
	A DIR	96	dd	2	3
	A EXT	B6	hh II	3	4
	A IND, X	A6	ff	2	4
	A IND, Y	18	A6	3	5
LDAB (opr)	B IMM	C6	ii	2	2
	B DIR	D6	dd	2	3
	B EXT	F6	hh II	3	4
	B IND, X	E6	ff	2	4
	B IND, Y	18	E6	3	5
LDD (opr)	IMM	CC	jj kk	3	3
	DIR	DC	dd	2	4
	EXT	FC	hh II	3	5
	IND, X	EC	ff	2	5
	IND, Y	18	EC	3	6
LDS (opr)	IMM	8E	jj kk	3	3
	DIR	9E	dd	2	4
	EXT	BE	hh II	3	5
	IND, X	AE	ff	2	5
	IND, Y	18	AE	3	6
LDX (opr)	IMM	CE	jj kk	3	3
	DIR	DE	dd	2	4
	EXT	FE	hh II	3	5
	IND, X	EE	ff	2	5
	IND, Y	CD	ff	3	6
LDY (opr)	IMM	18	CE	4	4
	DIR	18	DE	3	5
	EXT	18	FE	4	6
	IND, X	1A	EE	3	6
	IND, Y	18	EE	3	6
LSL (opr)	EXT	78	hh II	3	6
	IND, X	68	ff	2	6
	IND, Y	18	68	3	7
LSLA	A INH	48		1	2
LSLB	B INH	58		1	2
LSLD	INH	05		1	3
LSR (opr)	EXT	74	hh II	3	6
	IND, X	64	ff	2	6
	IND, Y	18	64	3	7
LSRA	A INH	44		1	2
LSRB	B INH	54		1	2
LSRD	INH	04		1	3
MUL	INH	3D		1	10
NEG (opr)	EXT	70	hh II	3	6
	IND, X	60	ff	2	6
	IND, Y	18	60	3	7
NEGA	A INH	40		1	2
NEGB	B INH	50		1	2
NOP	INH	01		1	2
ORAA (opr)	A IMM	8A	ii	2	2
	A DIR	9A	dd	2	3
	A EXT	BA	hh II	3	4
	A IND, X	AA	ff	2	4
	A IND, Y	18	AA	3	5
ORAB (opr)	B IMM	CA	ii	2	2
	B DIR	DA	dd	2	3
	B EXT	FA	hh II	3	4
	B IND, X	EA	ff	2	4
	B IND, Y	18	EA	3	5

MC68HC11A4

TABLE 3 – MC68HC11A4 INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES (CONTINUED)

Source Form(s)	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Machine Code Bytes (Total)	Execution Time (Cycles)
		Opcode	Operand(s)		
PSHA	A INH	36		1	3
PSHB	B INH	37		1	3
PSHX	INH	3C		1	4
PSHY	INH	18 3C		2	5
PULA	A INH	32		1	4
PULB	B INH	33		1	4
PULX	INH	38		1	5
PULY	INH	18 38		2	6
ROL (opr)	EXT IND, X IND, Y	79 69 18 69	hh II ff ff	3 2 3	6 6 7
ROLA	A INH	49		1	2
ROLB	B INH	59		1	2
ROR (opr)	EXT IND, X IND, Y	76 66 18 66	hh II ff ff	3 2 3	6 6 7
RORA	A INH	46		1	2
RORB	B INH	56		1	2
RTI	INH	3B		1	12
RTS	INH	39		1	5
SBA	INH	10		1	2
SBCA (opr)	A IMM A DIR A EXT A IND, X A IND, Y	82 92 B2 A2 18 A2	II dd hh II ff ff	2 2 3 2 3	2 3 4 4 5
SBCB (opr)	B IMM B DIR B EXT B IND, X B IND, Y	C2 D2 F2 E2 18 E2	II dd hh II ff ff	2 2 3 2 3	2 3 4 4 5
SEC	INH	0D		1	2
SEI	INH	0F		1	2
SEV	INH	0B		1	2
STAA (opr)	A DIR A EXT A IND, X A IND, Y	97 B7 A7 18 A7	dd hh II ff ff	2 3 2 3	3 4 4 5
STAB (opr)	B DIR B EXT B IND, X B IND, Y	D7 F7 E7 18 E7	dd hh II ff ff	2 3 2 3	3 4 4 5
STD (opr)	DIR EXT IND, X IND, Y	DD FD ED 18 ED	dd hh II ff ff	2 3 2 3	4 5 5 6

MC68HC11A4

TABLE 3 - MC68HC11A4 INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES (CONTINUED)

Source Form(s)	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Machine Code Bytes (Total)	Execution Time (Cycles)
		Opcode	Operand(s)		
STOP	INH	CF		1	2
STS (opr)	DIR	9F	dd	2	4
	EXT	BF	hh II	3	5
	IND, X	AF	ff	2	5
	IND, Y	18	AF ff	3	6
STX (opr)	DIR	DF	dd	2	4
	EXT	FF	hh II	3	5
	IND, X	EF	ff	2	5
	IND, Y	CD EF	ff	3	6
STY (opr)	DIR	18	DF dd	3	5
	EXT	18	FF hh II	4	6
	IND, X	1A	EF ff	3	6
	IND, Y	18	EF ff	3	6
SUBA (opr)	A IMM	80	ii	2	2
	A DIR	90	dd	2	3
	A EXT	B0	hh II	3	4
	A IND, X	A0	ff	2	4
	A IND, Y	18 A0	ff	3	5
SUBB (opr)	B IMM	C0	ii	2	2
	B DIR	D0	dd	2	3
	B EXT	F0	hh II	3	4
	B IND, X	E0	ff	2	4
	B IND, Y	18 E0	ff	3	5
SUBD (opr)	IMM	83	ii kk	3	4
	DIR	93	dd	2	5
	EXT	B3	hh II	3	6
	IND, X	A3	ff	2	6
	IND, Y	18 A3	ff	3	7
SWI	INH	3F		1	14
TAB	INH	16		1	2
TAP	INH	06		1	2
TBA	INH	17		1	2
TEST	INH	00		1	*
TPA	INH	07		1	2
TST (opr)	EXT	7D	hh II	3	6
	IND, X	6D	ff	2	6
	IND, Y	18 6D	ff	3	7
TSTA	A INH	4D		1	2
TSTB	B INH	5D		1	2
TSX	INH	30		1	3
TSY	INH	18 30		2	4
TXS	INH	35		1	3
TYS	INH	18 35		2	4
WAI	INH	3E		2	14 + n * *
XGDX	INH	8F		1	3
XGDY	INH	18 8F		2	4

* -infinity or until reset occurs

* -12 cycles are used beginning with the opcode fetch. A wait state is entered which remains in effect for an integer number of MPU E-clock cycles (n) until an interrupt is recognized. Finally two additional cycles are used to fetch the appropriate interrupt vector

dd = 8-bit direct address (\$0000-\$00FF) (high byte assumed to be \$00)

ff = 8-bit positive offset \$00 (0) to \$FF (255) (is added to index)

hh = high order byte of 16-bit extended address

ii = one byte of immediate data

jj = high order byte of 16-bit immediate data

kk = low order byte of 16-bit immediate data

ll = low order byte of 16-bit extended address

mm = 1 byte bit mask (set bits to be affected)

rr = signed relative offset \$80 (-128) to \$7F (+127) (offset relative to the address following the machine code offset byte)